A Minimum Energy Solution to Monocular Simultaneous Localization and Mapping

Andrea Alessandretti, António Pedro Aguiar, João Pedro Hespanha and Paolo Valigi

Abstract— **In this paper we propose an alternative solution to the Monocular Simultaneous Localization and Mapping (SLAM) problem. This approach uses a Minimum-Energy Observer for Systems with Perspective Outputs and provides an optimal solution. Contrarily to the most famous EKF-SLAM algorithm, this method yields a global solution and no linearization procedures are required. Furthermore, we show that the estimation error converges exponentially fast toward a neighborhood of zero, where this region increases gracefully with the magnitude of the input disturbance, output noise and initial camera position uncertainty.**

For practical purposes, we present also the filter in both continuous and discrete time form. Moreover, to show how to integrate a new landmark in the state estimation, a simple initialization procedure is presented. The filter performances are illustrated via simulations.

I. Introduction

The Simultaneous Localization and Mapping (SLAM) problem asks whether it is possible for a robot placed at an unknown location in an unknown environment to build a map of the environment while simultaneously determine its location within the map using only relative observation of the environment. The ability for a robot to localize itself and map the environment is a fundamental step toward the fully autonomous operation of a robotic system.

The SLAM problem has been widely analyzed and different solutions have been presented, see for example the work [10] that provide a comprehensive introduction to the topic. The main solutions are based on either nonlinear filtering or optimization techniques and an interesting comparison of the two can be found in [8]. The EKF-SLAM [11] and the FastSLAM [5], [4] are the two most famous filtering solutions based on the Extended Kalman Filter and Particle Filter, respectively.

A key difficulty with the classical EKF-SLAM approach stems from the nonlinearity of the motion and observation models. The consistency of the EKF-SLAM is analyzed in [1] and [2], and eventual inconsistency of the algorithm has been proved especially for large maps. In spite of that, many successful applications have been carried out thanks to a variety of methods used to reduce the approximation errors. One of the methods that contributed to the SLAM efficiency and effectiveness are sub-mapping techniques, see for instance [6].

When only a single camera sensor is used we are dealing with Monocular SLAM where the effects of the nonlinearity in the observation model is particularly significant. In fact, if the landmarks estimates are far from the real value, the linearization error can be great. A wide literature addresses the problem of feature initialization, see for instance [3] and [9]. An inverse depth landmark parameterization is presented in [13] which, along with other advantages, reduces the observation model nonlinearity.

The main results of this work is the introduction of a new approach based on Minimum-Energy estimation theory for systems with perspective outputs [7] that solves the Monocular SLAM problem. The result is an optimal filtering solution where, in absence of input disturbances, output noise and with no uncertainty on the initial robot pose, the estimation error converges exponentially to zero. In case of initial camera pose uncertainty or when input disturbance or output noise are present, the estimation error converges exponentially to a bounded region around zero, where the width of this region is proportional to the magnitude of these disturbances. We highlight that, under the assumption of the model presented and given that no linearization error is introduced, we provide a global and optimal solution against the local solution of the EKF-SLAM. This implies, for instance, that no special landmarks initialization procedure is required, and the landmark position estimate will converge toward the real value independently from how it is initialized.

The remainder of this paper is organized as follows. Section II states the SLAM problem. In Section III we show how to write the SLAM problem as an estimation problem of system with perspective outputs and we present the observer equations in both continuous and discrete time. We close the Section with the filter convergence propriety. The Section IV is dedicated to features initialization. Using simulation, in Section V we show the filter behavior in different scenarios. In Section VI we provide some final conclusions.

II. Problem Statement

Consider a coordinate frame $\{\mathrm{C}\}$ attached to a camera, which moves with respect to an inertial frame $\{\mathrm{W}\}$. Let $(p_{WC}, R_{WC}) \in \mathrm{SE}(3)$ be the configuration of the camera frame $\{\mathrm{C}\}$, where SE(3) is the Cartesian product of $\mathbb{R}^3$ with the group SO(3) of 3×3 rotation matrices. Given a set of N

This work was supported in part by projects CONAV (PTDC/EEA-CRO/113820/2009), Co3-AUVs (EU FP7 under grant agreement No. 231378), FCT (ISR/IST plurianual funding) through the PIDDAC Program funds, the CMU-Portugal program and NSF Grant CNS-0720842.

Andrea Alessandretti and António Pedro Aguiar are with the Institute for Systems and Robotics, Instituto Superior Técnico, Technical University of Lisbon, Portugal. andrea.alessandretti@gmail.com - pedro@isr.ist.utl.pt

João Pedro Hespanha is University of California, Santa Barbara (UCSB), United States hespanha@ece.ucsb.edu

Paolo Valigi is with the Dipartimento di Ingegneria Elettronica e dell'Informazione, Università degli Studi di Perugia, Italy paolo.valigi@diei.unipg.it

landmarks, let $l_i^W \in \mathbb{R}^3$ and $l_i^C \in \mathbb{R}^3$ with $i \in \{1, \ldots, N\}$ denote the coordinates of the ith landmark in world frame and camera frame, respectively. Then we have

$$l_i^W \quad = \quad p_{WC} + R_{WC} l_i^C \tag{1}$$

Now, let $(v^C, \Omega^C) \in \mathrm{se}(3)$ be the twist that defines the velocity of the frame {C} with respect to {W}, expressed in the frame {C}. The symbol se(3) represents the Cartesian product of $\mathbb{R}^3$ with the space so(3) of 3×3 skew-symmetric matrix. The following holds

$$v^C = R_{WC}^T p_{\dot{W}C}, \qquad \Omega^C = R_{WC}^T \dot{R}_{WC}, \tag{2}$$

where $v^C \in \mathbb{R}^3$ is the linear velocity of the camera expressed in {C} and Ω^C is defined by the angular velocity vector $\omega^C = [\omega_1, \omega_2, \omega_3]^T$ as follows

$$\Omega^C = \begin{pmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{pmatrix}.$$

Using a bearing only sensor we observe the projection of the generic point $l_i^C = [l_{i,1}^C, l_{i,2}^C, l_{i,3}^C]$ in the camera sensor. A normalized version of the observation is the homogeneous image coordinate that follows,

$$y_i = \begin{pmatrix} \frac{l_{i,1}^C}{l_{i,3}^C} & \frac{l_{i,2}^C}{l_{i,3}^C} & 1 \end{pmatrix}^T.$$

Our goal is to estimate iteratively, as measurements are arriving, the camera and landmarks final positions.

III. Minimum Energy SLAM

In this section the ME-SLAM approach is presented. We start by defining a generic system with perspective outputs, then we show that it is possible to write the Monocular SLAM problem as state estimation problem of such system. We conclude with the filter expressions in both continuous and discrete form.

A. System with perspective outputs

A state affine system with multiple perspective outputs is of the form

$$\begin{aligned} \dot{x} \quad &= \quad A(u)x + b(u) + G(u)d \qquad (3) \\ \alpha_j y_j \quad &= \quad C_j(u)x + d_j(u) + n_j \qquad (4) \\ & \quad j \in \mathcal{J} := \{1, 2, \ldots, N\} \end{aligned}$$

where $x \in \mathbb{R}^n$ is the state vector, $u \in \mathbb{R}^{n_u}$ is the control input, $y_j \in \mathbb{R}^{m_j}$ is the jth perspective output, $d \in \mathbb{R}^{n_d}$ an input disturbance that cannot be measured, $n_j \in \mathbb{R}^{m_j}$ measurement noise affecting the jth output, $A : \mathbb{R}^{n_u} \rightarrow \mathbb{R}^{n \times n}$, $b : \mathbb{R}^{n_u} \rightarrow \mathbb{R}^n$, $G : \mathbb{R}^{n_u} \rightarrow \mathbb{R}^{n \times n_d}$, $C_j : \mathbb{R}^{n_u} \rightarrow \mathbb{R}^{m_j \times n}$, $d_j : \mathbb{R}^{n_u} \rightarrow \mathbb{R}^{m_j}$. The right-hand-side of (4) is assumed to be always non zero, and the initial condition $x(0)$, the signal d and n_j are all assumed to be deterministic but unknown. Each $\alpha_j \in \mathbb{R}$, $j \in \mathcal{J}$ denotes a scalar determined by a normalization constraint

$$\|y_j\| = 1 \quad \text{or} \quad v_j^T y_j = 1. \tag{5}$$

The scalar α_j is *unknown* and contains the information about the landmark depth. The constraint (5) shows that the observations y_i, perspective observations, do not carry any information in their module but only in their direction. This is exactly what happen when a landmark position is projected in the camera sensor. Moreover, we assume that measurements are available at the sampling times t_i, $i \in \mathcal{I}_o \subseteq \mathcal{I} := \{0, \ldots, k\}$ with $t_0 := 0 \leq t_1 \leq \cdots \leq t_k$ and that only a subset $\mathcal{J}_i$ of the N measurements is available at time t_i.

B. From SLAM to a system with perspective outputs

To model the observation from a bearing-only sensor as perspective outputs, consider the generic ith landmark. The image coordinates of this landmark (which are the only observation measure for the estimation problem) are given by y_i that satisfies

$$\alpha_i y_i = l_i^C, \tag{6}$$

where α_i is an *unknown* scalar containing the information about the landmark depth. From (1) it follows that

$$\begin{aligned} l_i^C \quad &= \quad R_{WC}^T l_i^W - R_{WC}^T p_{WC} \\ &= \quad l_i - p \end{aligned}$$

where we defined $l_i := R_{WC}^T l_i^W$ and $p := R_{WC}^T p_{WC}$.

In order to obtain a linear output function, we consider the following state vector

$$x = (p, l_1, \ldots, l_N)^T \in \mathbb{R}^{3N+3}.$$

This step is important because it motivates our features and camera pose representation in the state vector. In fact, using inertial coordinates we would have a nonlinear perspective output equation, while using this representation we have

$$\alpha_i y_i = l_i - p,$$

which is linear in the state. We remark that we are not using a linear output in the classical sense, but we are using a linear *perspective* output.

Now we proceed with the analysis of the kinematic equations. Using (2) and assuming static landmarks (i.e. $\dot{l}_1^W = 0$) we have

$$\begin{aligned} \dot{p} \quad &= \quad \dot{R}_{WC}^T p_{WC} + R_{WC}^T \dot{p}_{WC} \\ &= \quad -\Omega_{WC}^C R_{WC}^T p_{WC} + v^C \\ &= \quad -\Omega_{WC}^C p + v^C \end{aligned}$$

and

$$\begin{aligned} \dot{l}_i \quad &= \quad \dot{R}_{WC}^T l_i^W + R_{WC}^T \dot{l}_i^W \\ &= \quad -\Omega_{WC}^C R_{WC}^T l_i^W \\ &= \quad -\Omega_{WC}^C l_i. \end{aligned}$$

Defining the control $u := \{v^C, \omega^C\}$ and using the Kronecker product $\otimes$, we can write our system in the form (3)-(4)

$$A(u) = I_{(N+1)\times(N+1)} \otimes -\Omega^C(\omega^C) \in \mathbb{R}^{(3N+3)\times(3N+3)}$$

$$b(u) = \begin{pmatrix} v_C \\ 0_{3N\times 1} \end{pmatrix} \in \mathbb{R}^{3N+3}$$

$$\begin{pmatrix} C_1 \\ \vdots \\ C_N \end{pmatrix} = \begin{pmatrix} -I_{3\times 3} \otimes 1_{N\times 1} & I_{3N\times 3N} \end{pmatrix} \in \mathbb{R}^{3N\times 3N+3}$$

where I and 1 are identity and ones matrixes, respectively, with dimension specified by the subscript.

For practical purposes we also present a discretized version of the state equation. Consider to receive input control at time $t_i, i \in \mathcal{I}$. We remark that the observation arrival identified by $\mathcal{I}_o \subseteq \mathcal{I}$ can be a subset of the control arrival time. Assuming that the control inputs are constant for all the interval of time $\Delta t = t_{i+1} - t_i$ we have

$$\begin{aligned} p(t_i + \Delta t) &= e^{-\Omega^C_{WC}\Delta t} p(t_i) \\ &\quad + v^C \int_{t_i}^{t_i+\Delta t} e^{-\Omega(t+\Delta t-\tau)} d\tau \\ &\approx e^{-\Omega^C_{WC}\Delta t} p(t_i) + v^C e^{-\Omega^C_{WC}\Delta t}\Delta t \\ &= R(-\omega^C, \Delta t) p(t_i) + R(-\omega^C, \Delta t) v^C \Delta t \end{aligned}$$

where $R(-\omega^C, \Delta t) = e^{-\Omega^C_{WC}\Delta t} \in SO(3)$ is the rotation matrix associated to the rotation for a time Δt about the axes ω^C with angular velocity $-|\omega^C|$. In the same way, for the generic ith landmark we obtain

$$\begin{aligned} l_i(t_i + \Delta t) &= e^{-\Omega^C_{WC}\Delta t} l_i(t_i) \\ &= R(-\omega^C, \Delta t) l_i(t_i). \end{aligned}$$

The discrete time model is then given by

$$x(t_{i+1}) = A_i x(t_k) + b_i + G_i d_i$$

where

$$A_i = I_{(N+1)\times(N+1)} \otimes R(-\omega^C, \Delta t) \in \mathbb{R}^{(3N+3)\times(3N+3)}$$

$$b_i = \begin{pmatrix} R(-\omega^C, \Delta t) v^C \Delta t \\ 0_{(3N)\times 1} \end{pmatrix} \qquad \in \mathbb{R}^{3N+3}$$

$G : \mathbb{R}^{n_u} \to \mathbb{R}^{n \times n_d}$ and $d_i \in \mathbb{R}^{n_d}$ is the discrete disturbance that cannot be measured.

C. Observer equations

In the previous section we showed how to convert the Monocular SLAM problem to a estimation problem of a system with perspective outputs. To estimate the state of this class of systems we use the filter presented in [7]. For practical purposes, within this work we also present the discrete time version.

We let the reader note that we could redefine the output equation as $y_i = (l_i - p)/|(l_i - p)|$, and then apply EKF. However, due to linearization, this would lead to a local solution, meaning that the filter convergence would be guaranteed only for an initial landmark position guess sufficiently close to the true value.

With our approach, given an input u defined on an interval $[0, t)$, and a measured output $y_j(t_i)$, $j \in \mathcal{J}_i$ with $i \in \mathcal{I}_o$, we obtain the state estimate $\hat{x}(t)$ at time t defined by

$$\hat{x}(t) := \arg \min_{z \in \mathbb{R}^n} J(z, t) \tag{7}$$

where

$$\begin{aligned} J(z;t) := & \min_{\substack{d:[0,t), \bar{n}_j(t_i), \alpha_{j_i} \\ i=0,1,\ldots,k}} \{ (x(0) - \hat{x_0})^T P_0^{-1} (x(0) - \hat{x_0}) \\ & + \int_0^t \|d(\rho)\|^2 d\rho + \sum_{i=0}^k \sum_{j \in \mathcal{J}} \|n_j(t_i)\|^2 : \\ & x(t) = z, \dot{x} = A(u)x + b(u) + G(u)d, \\ & \alpha_{j_i} y_j(t_i) = C_j x_{t_i} + d_j(u) + n_j(t_i) \} \end{aligned} \tag{8}$$

and $P_0 > 0$, $\hat{x}_0$ encode *a priori* information about the state.

Note that if we pose no restrictions on the state disturbance and output noise of (3)-(4), a measured sequence of observations could correspond to any state solution. The solution (7) corresponds to the state solution that needs less amount of disturbance and noise to be explained. Notice also that in general the solution of this minimum energy formulation for a general nonlinear system leads to an infinite dimensional observer, whose state evolves according to a first order nonlinear PDE of Hamilton-Jacobi type, driven by the observation. However, for the case of a linear system we obtain the Kalman filter, and for perspective systems we can also obtain an exact close form solution that is filtering-like and iterative.

The filter equation that solves (7)-(8) are the following
- for $t_i \leqslant t < t_{i+1},\ i \in \mathcal{I}_o$

$$\begin{aligned} \dot{P} &= A(u)P(t) + P(t)A(u)^T + G(u)G^T(u), & P(t_i) &= P_i \\ \dot{\hat{x}} &= A(u)\hat{x}(t) + b(u), & \hat{x}(t_i) &= \hat{x}_i \end{aligned} \tag{9}$$

- at $t = t_i,\ \in \mathcal{I}_o$

$$\begin{aligned} P(t_i) &= (P(t_i^-)^{-1} + W(t_i))^{-1} \\ \hat{x}(t_i) &= \hat{x}(t_i^-) - P(t_i^-)\left(W(t_i)\hat{x}(t_i^-) + w(t_i)\right. \end{aligned} \tag{10}$$

where

$$\begin{aligned} W(t_i) &:= \sum_{j \in \mathcal{J}_k \subseteq \mathcal{J}} C_j^T(u) \left(I - \frac{y_j(t_i) y_j(t_i)^T}{\left\| y_j(t_i) \right\|^2} \right) C_j(u) \\ w(t_i) &:= \sum_{j \in \mathcal{J}_k \subseteq \mathcal{J}} C_j^T(u) \left(I - \frac{y_j(t_i) y_j(t_i)^T}{\left\| y_j(t_i) \right\|^2} \right) \bar{d}_j(u). \end{aligned}$$

If we are interested on working with the discrete time model, we can replace the continuous time Riccati equation (9) with the discrete time version, obtaining
- at $t = t_{i+1},\ i \in \mathcal{I}$

$$\begin{aligned} P(t_{i+1}) &= A_i P(t_i) A_i^T + G_i G_i^T \\ \hat{x}(t_{i+1}) &= A_i \hat{x}(t_i) + b_i \end{aligned} \tag{11}$$

- at $t = t_{i+1},\ i \in \mathcal{I}_o$

$$\begin{aligned} P(t_{i+1}) &= (P(t_{i+1}^-)^{-1} + W(t_{i+1}))^{-1} \\ \hat{x}(t_{i+1}) &= \hat{x}(t_{i+1}^-) - P(t_{i+1}^-) W(t_{i+1}) \hat{x}(t_{i+1}^-) \end{aligned} \tag{12}$$

where equations (11) and (12) can be considered as the counterpart of the prediction step and update step of the Kalman filter, respectively.

As can be seen from (8), the initial value P_0 of P(t) reflects our confidence on the initial estimate $\hat{x}_0$ of $x(0)$, e.g. a large value of P_0 strongly penalizes any deviation of the state from our “a-priori” guess $\hat{x}_0$. Because of this, in fact, we can interpret the initial value of P(t) as the covariance matrix that reflects our certainty on our initial guess for the state.

D. Convergence

Similarly to the minimum energy estimator for systems with perspective outputs, under suitable observability assumptions (see Theorem 3 of [7]), the state estimation error converges exponentially fast to a neighborhood. Furthermore, the estimation error degrades gracefully with the increasing of the magnitude of the output noise and state disturbance. For the specific case of the SLAM problem the estimation error also depends on the uncertainty of the initial camera pose. In Appendix we provide a lower bound on the covariance matrix associated with any single landmark estimate as a function of the initial covariance in the camera position.

IV. Features initialization

The theory behind the convergence of the ME-SLAM outlined above guarantees global convergence for any initialization of the filter. However, in practice, the performance of the filter can be significantly improved by suitably feature initialization, which corresponds to choosing appropriate initial conditions to (9).

In the literature we find several accurate initialization techniques for bearing-only sensor. Since our system does not require special accuracy we propose an intuitive method and show how to update the state vector and covariance matrices.

Let $y_n = [y_1, y_2, 1]^T$ be the homogeneous image coordinate associated to the landmark that we would like to initialize, which has camera frame coordinate l_n^C. The vector $[y_1, y_2]^T$ is corrupted by additive sensor noise, with known covariance matrix $R \in \mathbb{R}^{2\times 2}$. Assume to have a priori information about the distribution of the depth of the observed landmarks, namely its first and second statistical moment, $\hat{\rho}$ and σ_ρ^2. For instance, during indoor exploration we can exclude the possibility to observe landmarks 50 meters far from the camera, and it is reasonable to assume some distribution over closer distances.

A simple approach to feature initialization consists in defining an *initilization function* $g : \mathbb{R}^3 \to \mathbb{R}^3$ as

$$l_n^C = g(y_1, y_2, \rho) = \begin{pmatrix} \rho y_1 & \rho y_2 & \rho \end{pmatrix}^{\mathrm{T}}$$

Since $g(.)$ is a function of random variables, than l_n^C is also a random variable and a linear estimate of it is given by the following first and second statistical moments

$$\begin{aligned} \hat{l}_n^C &= g(y_1, y_2, \hat{\rho}) = \begin{pmatrix} \hat{\rho} y_1 & \hat{\rho} y_2 & \hat{\rho} \end{pmatrix}^{\mathrm{T}}, \qquad (13) \\ P_{l_n} &= \nabla_g \begin{pmatrix} R & 0 \\ 0 & \sigma_\rho^2 \end{pmatrix} \nabla_g^T, \qquad (14) \end{aligned}$$

where ∇_g is the Jacobian of $g(.)$ evaluated at $\hat{y}$ and $\hat{\rho}$

$$\nabla_g = \begin{pmatrix} \hat{\rho} & 0 & y_1 \\ 0 & \hat{\rho} & y_2 \\ 0 & 0 & 1 \end{pmatrix}.$$

Using this initialization function we set the initial estimate of the new landmark on a position coherent with the observation y_n on a plane in front of the camera, parallel to the camera sensor and distant $\hat{\rho}$ from the optical camera center. Thus, the initialization of a new landmark is

$$x = \begin{pmatrix} x \\ \hat{l}_n^C + p \end{pmatrix} \quad , \quad P = \begin{pmatrix} P & 0 \\ 0 & P_{l_n} + P_p \end{pmatrix} \qquad (15)$$

where P_p is the covariance matrix relative to the vector p and 0 is a zero matrix with appropriate dimensions. After (15) an update using the current observations is desirable to correlate the new landmark with the other state components.

If we wish to remove some landmarks form the state, it is enough to delete the component of the state vector and covariance matrix relative to those landmarks.

A. Pseudocode

In this section we present a pseudocode that summarizes the overall system procedures.

Algorithm 1 ME-SLAM

```
$\mathcal{L}_n \leftarrow \emptyset$
$\mathcal{L}_m \leftarrow \emptyset$
$\mathcal{L} \leftarrow \emptyset$
$\mathcal{L}_s \leftarrow \emptyset$
$(x, P) \leftarrow$ system_initialization
for all $i \in \mathcal{I}$ do
    if $i \in \mathcal{I}_o$ then
        $img \leftarrow$ get_image
        $\mathcal{L} \leftarrow$ feature_extraction($img$)
        $(\mathcal{L}_n, \mathcal{L}_m) \leftarrow$ feature_mathing($\mathcal{L}, \mathcal{L}_s$)
        $\mathcal{L}_s \leftarrow$ initialize_landmarks($\mathcal{L}_s, \mathcal{L}_n$)
        $(x, P) \leftarrow$ update($x, P, \mathcal{L}$)
        $(x, P, \mathcal{L}_s) \leftarrow$ feature_selection($x, P, \mathcal{L}_s$)
        $(x, P) \leftarrow$ prediction($x, P$)
    else
        $(x, P) \leftarrow$ prediction($x, P$)
    end if
end for
```

In the pseudocode, $\mathcal{L}$ is the set of features extracted from an image, each of its elements carries the homogeneous image coordinate y_i of the observed landmark and a descriptor d_i used to identify the landmark in different images. With $\mathcal{L}_m$ and $\mathcal{L}_n$ we identify the subset of matched features and new features respectively, $\mathcal{L}_n \cup \mathcal{L}_m = \mathcal{L}$. These sets are obtained by the function `feature_mathing` using the set $\mathcal{L}_s$ of descriptors associated to the landmarks in the current state vector and comparing them with the descriptors of the landmark just observed. The function `initialize_landmarks` initialize the new landmarks and refers to the set of equation (13) (14) and (15). The estimation is updated with the function `update` using the observation. This step refers to the equations (12). The function `feature_selection` is used to perform a selection of robust landmarks. For instance, it is reasonable to discard a landmark if only one observation has been collected along a sequence of consecutive camera images. These extra-information needed for the selection are considered part of $\mathcal{L}_s$. Also information from x, P can be used to support this operation. Finally the function `prediction` predicts the future state using (11).

V. Simulation Results

In this section we show the filter performance via simulation where the camera moves inside a room along the walls and takes observations of the landmarks placed over the walls. The simulation setting is as follows. The initial camera position is $\begin{pmatrix}0 & 0 & 0\end{pmatrix}^{\mathrm{T}}$ and faces the wall defined by the landmarks displaced between $\begin{pmatrix}-6 & 2 & 0\end{pmatrix}^{\mathrm{T}}$ and $\begin{pmatrix}6 & 2 & 0\end{pmatrix}^{\mathrm{T}}$ in the frame of Fig. 1. Then, the camera turns right and starts moving along the corridor at velocity of 1 m/s. It takes observations of landmarks in front of the camera that are closer than 10 m. Whenever a landmarks is observed for the first time, a state component is initialized assuming an initial landmark depth of 0 m and a $\sigma_\rho = 20$m. We use an unreasonable initial guess of the depth to show the filter behavior in case of significant initial estimation error. Moreover, we simulated additive zero mean noise on the camera state equation and output equation, both with correlation matrix of $0.001 I_{3\times3}$, where $I_{3\times3}$ it a 3×3 identity matrix.

In Fig. 1 it is worth to notice that two of the landmarks close to the origin have a wide covariance. This is because the camera started its exploration turning right and the filter did not experience enough parallax to reduce the two landmarks covariances. Similarly happens for the landmark in front of the camera at time t=35s.

This phenomena is observable also from the bottom plot of Fig. 3 where the trace of the covariance matrices relative to the position of landmarks and camera are displayed. We see that the covariances of these two landmarks only converge to a value around zero when they are observed again at the end of the loop, around time t = 42s and t = 46s. The effect of this on the estimation error is visible from the top plot, where the estimation error is shown. The spikes correspond to the initialization times of the landmarks, after which the covariance matrices and the estimation errors converge to small values.

VI. Conclusions

We presented an alternative solution to the Monocular SLAM, by rewriting the problem as a state affine system with multiple perspective outputs (3)-(4). Using this formulation we avoid linearization, main cause of EKF-SLAM divergence, and we provide a global solution to the Monocular SLAM problem against the local solution of the EKF-SLAM.

Using a Minimum-Energy observer we can guarantee the estimation error to converge exponential to zero in

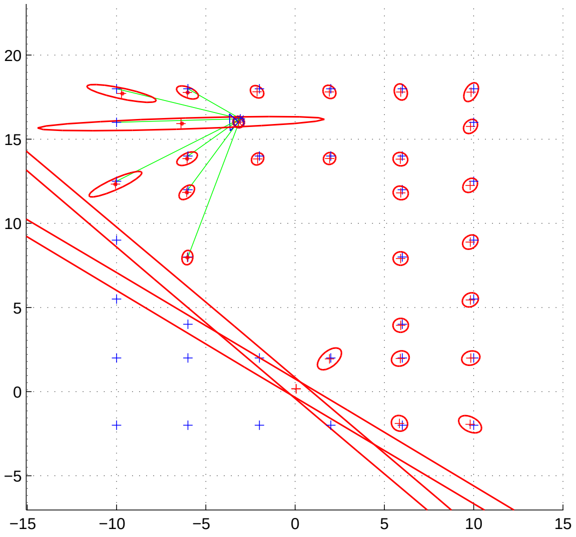

Fig. 1. Top view of the 3D scenario at time t = 35 s. The blue crosses represent the true landmarks position, the means and the covariances of their estimates are represented by the red crosses and the red ellipses, respectively. The red asterisk and the red ellipse stand for the estimate of the camera position. A draw of the camera shows the heading and the green line identify the observed landmarks. The scale is in meters.

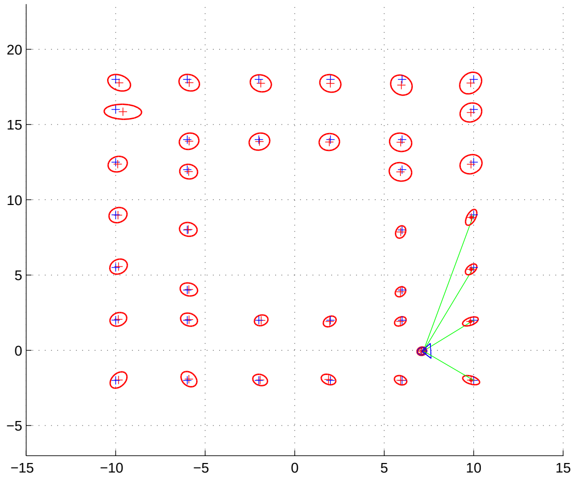

Fig. 2. Top view of the 3D scenario at time t = 70.4 s. The blue crosses represent the true landmarks position, the means and the covariances of their estimates are represented by the red crosses and the red ellipses, respectively. The red asterisk and the red ellipse stand for the estimate of the camera position. A draw of the camera shows the heading and the green line identify the observed landmarks. The scale is in meters.

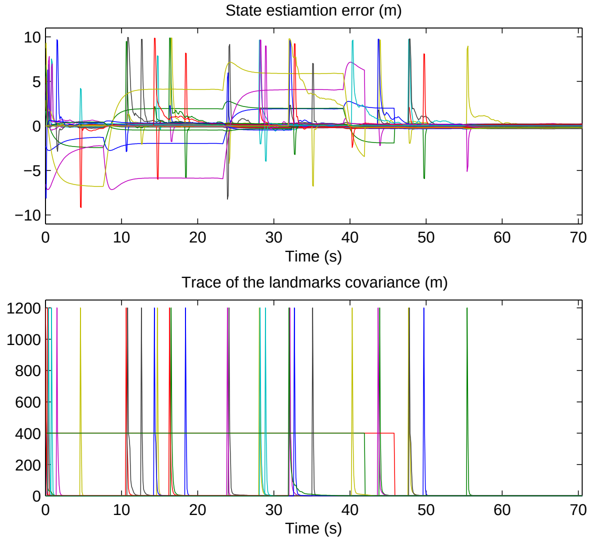

Fig. 3. The plot on the top shows the state estimation error. The plot on the bottom shows the trace of the covariance matrices of each landmark separately.

absence of input disturbance, output noise and initial camera pose uncertainty. When input disturbance, output noise and initial robot pose uncertainty are present, the estimation error degrades gracefully with the magnitude of the input disturbance, output noise and initial camera pose uncertainty.

We presented the system and the observer equation in both continuous form, (9)-(10), and discrete form, (11)-(12). This last manifests the two steps Prediction-Update that is particularly useful for practical implementation.

Given the dynamic nature of the SLAM problem, where the number of landmark that we estimate grows with time, in Section IV we presented a recursive initialization procedure, (13), (14) and (15), suitable for our state vector.

Finally, the filter behavior has been shown via simulation. Here the traces of the landmarks position uncertainty converge toward a value around zero as expected.

Appendix

Using a similar approach of the one used in [15] for the EKF SLAM, in this section we show that the lower bound of the covariance matrix is determined by the initial uncertainty on the camera pose.

For sake of simplicity we consider a single landmark. In order to analyze the lower bound, we assume to use observation from a stationary camera (i.e. $P(t_i^-) = P(t_{i-1})$). Then, from (12) we have

$$\begin{pmatrix} P_p^{-1}(t_i) & P_{pm}^{-1}(t_i) \\ P_{mp}^{-1}(t_i) & P_m^{-1}(t_i) \end{pmatrix} = \begin{pmatrix} P_p^{-1}(t_{i-1}) & P_{pm}^{-1}(t_{i-1}) \\ P_{mp}^{-1}(t_{i-1}) & P_m^{-1}(t_{i-1}) \end{pmatrix} + \begin{pmatrix} M(t_i) & -M(t_i) \\ -M(t_i) & M(t_i) \end{pmatrix} \tag{16}$$

where $M(t_i) = I - y(t_i)y(t_i)^T/\|y(t_i)\|^2$ is a positive semidefinite matrix, P_p the covariance matrix of the camera position, P_m the covariance matrix of the landmark and P_{mp} and P_{pm} cross covariance matrices. If there is no information on the landmark position at time $t = 0$ we have

$$P^{-1}(0) = \begin{pmatrix} P_p^{-1}(0) & 0 \\ 0 & 0 \end{pmatrix}. \tag{17}$$

Then, from (16) and (17) we obtain

$$\begin{pmatrix} P_p^{-1}(t_i) & P_{pm}^{-1}(t_i) \\ P_{mp}^{-1}(t_i) & P_m^{-1}(t_i) \end{pmatrix} = \begin{pmatrix} P_p^{-1}(0) + \bar{M}(t_i) & -\bar{M}(t_i) \\ -\bar{M}(t_i) & \bar{M}(t_i) \end{pmatrix}$$

with $\bar{M}(t_i) = \sum_{k \leq i, k \in \mathcal{I}_o} M(t_k)$. Invoking the matrix inversion lemma for partitioned matrices, we can conclude that

$$\lim_{i \to \infty} \begin{pmatrix} P_p(t_i) & P_{pm}(t_i) \\ P_{mp}(t_i) & P_m(t_i) \end{pmatrix} \geq \begin{pmatrix} P_p(0) & P_p(0) \\ P_p(0) & P_p(0) \end{pmatrix}.$$

References

[1] Tim Bailey, Juan Niet, Jose Guivant, Michael Stevens and Eduardo Nebot, “Consistency of the EKF-SLAM algorithm,” *in IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3562–3568 2006.

[2] Shoudong Huang and Gamini Dissanayake, “Convergence and consistency analysis for Extended Kalman Filter based SLAM,” *in IEEE Transactions on robotics* , pages 1036–1049,2007.

[3] Rodrigo Munguia and Antoni Grau,“Delayed Feature Initialization for Inverse Depth Monocular SLAM,” *in European Conference on Mobile Robots*, pages 1–6, 2007.

[4] Michael Montemerlo, Sebastian Thrun, Daphne Koller and Ben Wegbreit, “FastSLAM 2.0: An improved particle filtering algorithm for simultaneous localization and mapping that provably converges,” *in International Joint Conference on Artificial Intelligence*,pages 1151–1156,2003.

[5] Michael Montemerlo, Sebastian Thrun, Daphne Koller, and Ben Wegbreit, “FastSLAM : A Factored Solution to the Simultaneous Localization and Mapping Problem,” *in Proceedings of the National conference on Artificial Intelligence*, pages 593–598, 2002.

[6] P. Pinies and Juan D. Tardos, “Large-Scale SLAM Building Conditionally Independent Local Maps: Application to Monocular Vision,” *in IEEE Transactions on Robotics*,pages 1094–1106, 2008.

[7] António Pedro Aguiar and João Pedro Hespanha, “Minimum-Energy State Estimation for Systems with Perspective Outputs,” *in IEEE Transaction on Automatic Control*,pages 226—-241, 2006.

[8] Hauke Strasdat, J.M.M. Montiel and Andrew J. Davison, “Real-Time Monocular SLAM: Why Filter?,” *in International Conference on Robotics and Automation*, 2010.

[9] Andrew J. Davison, “Real-Time Simultaneous Localisation and Mapping with a Single Camera,” *in Proceedings International Conference on Computer Vision*, pages 1403–1410, 2003.

[10] Tim Bailey and Huge Durrant-Whyte,“Simultaneous Localization And Mapping (SLAM): part II, ”*in IEEE Robotics & Automation Magazine*, pages 108–117, 2006.

[11] Huge Durrant-Whyte and Tim Bailey, “Simultaneous Localization And Mapping: part I,” *in IEEE Robotics & Automation Magazine*, pages 99–108, 2006.

[12] C. Cadena and J. Neira, “SLAM in O(logn) with the Combined Kalman-Information Filter, ” *in Robotics and Autonomous Systems*, pages 1207–1219, 2010.

[13] J.M.M. Montiel, Javier Civera, and Andrew J. Davison, “Unified inverse depth parameterization for Monocular SLAM,” *in Proceedings of Robotics: Science and Systems*, Philadelphia, USA, August 2006.

[14] Michael Montemerlo, and Sebastian Thrun, “Simultaneous Localization And Mapping with unknown data association using FastSLAM,” *in Robotics and Automation*, 2003.

[15] M. W. M. Gamini Dissanayake, Paul Newman, Steven Clark, Hugh F. Durrant-Whyte and M. Csorba, “A solution to the simultaneous localization and map building (SLAM) problem.” *IEEE Transactions on Robotics and Automation, 17(3), 229-241,* 2001