

Article

Concurrent Initialization for Bearing-Only SLAM

Rodrigo Munguía and Antoni Grau *

Automatic Control Department, Technical University of Catalonia, c/ Pau Gargallo, 5 E-08028 Barcelona, Spain; E-Mail: rodrigo.munguia@upc.edu

* Author to whom correspondence should be addressed; E-Mail: antoni.grau@upc.edu; Tel.: +34-93-401-6975; Fax: +34-93-401-7045.

Received: 18 December 2009; in revised form: 15 February 2010 / Accepted: 24 February 2010 / Published: 1 March 2010

Abstract: Simultaneous Localization and Mapping (SLAM) is perhaps the most fundamental problem to solve in robotics in order to build truly autonomous mobile robots. The sensors have a large impact on the algorithm used for SLAM. Early SLAM approaches focused on the use of range sensors as sonar rings or lasers. However, cameras have become more and more used, because they yield a lot of information and are well adapted for embedded systems: they are light, cheap and power saving. Unlike range sensors which provide range and angular information, a camera is a projective sensor which measures the bearing of images features. Therefore depth information (range) cannot be obtained in a single step. This fact has propitiated the emergence of a new family of SLAM algorithms: the Bearing-Only SLAM methods, which mainly rely in especial techniques for features system-initialization in order to enable the use of bearing sensors (as cameras) in SLAM systems. In this work a novel and robust method, called *Concurrent Initialization*, is presented which is inspired by having the complementary advantages of the Undelayed and Delayed methods that represent the most common approaches for addressing the problem. The key is to use concurrently two kinds of feature representations for both undelayed and delayed stages of the estimation. The simulations results show that the proposed method surpasses the performance of previous schemes.

Keywords: bearing-sensor; SLAM; robotics

1. Introduction

Simultaneous Localization and Mapping (SLAM) is perhaps the most fundamental problem to solve in robotics in order to build truly autonomous mobile robots. SLAM treats of the way how a mobile robot can operate in an *a priori* unknown environment using only onboard sensors to simultaneously building a map of its surroundings which uses to track its position.

The robot's sensors have a large impact on the algorithm used for SLAM. Early SLAM approaches focused on the use of range sensors as sonar rings or lasers e.g., [1]. Nevertheless there are some disadvantages with the use of range sensors in SLAM: correspondence or data association is difficult; they are expensive and some of them are limited to 2D maps and computational overhead due to large number of features (see [2,3] for a complete review).

The aforementioned issues have propitiated that recent work moves towards the use of cameras as the primary sensing modality. Cameras have become more and more interesting for the robotic research community, because they yield a lot of information for data association, although this problem remains latent. Cameras are well adapted for embedded systems: they are light, cheap and power saving. Using vision, a robot can localize itself using common objects as landmarks.

On the other hand, while range sensors (*i.e.*, laser) provide range and angular information, a camera is a projective sensor which measures the bearing of images features. Therefore depth information (range) cannot be obtained in a single frame. This fact has propitiated the emergence of a new family of SLAM methods: The Bearing-Only SLAM methods, which mainly rely in especial techniques for features system-initialization in order to enable the use of bearing sensors (as cameras) in SLAM systems.

In this context, a camera connected to a computer becomes a position sensor which could be applied to different fields such as robotics (motion estimation for generally moving robots humanoids), wearable robotics (motion estimation for camera equipped devices worn by humans), tele-presence (head motion estimation using an outward-looking camera), or television (camera motion estimation for live augmented reality) [4].

Usually the Bearing-Only SLAM has been associated with vision-based SLAM systems, possibly because cameras are by far the most popular bearing sensor used in robotics. In that sense, the use of alternative bearing sensors (*i.e.*, auditory sensing) for performing SLAM has been much less explored. Nevertheless in an authors' previous work [5], a Sound-Based SLAM system is proposed where sound sources are used as map features and thus showing the viability on the inclusion of the hearing sense in SLAM and the use of alternative bearing sensors.

In recent years several important improvements and variants to this kind of methods have appeared [6,7]. Also different schemes for increasing the number of features managed into the map have appeared [8]. Nevertheless the initialization process of new features is still the most important problem for addressing in Bearing-Only SLAM in order to improve the robustness.

In this work a novel and robust method called *Concurrent Initialization* is presented which is inspired by having the complementary advantages of the Undelayed and Delayed methods, which represent the most common approaches for addressing the problem of initializing new features in bearing-only SLAM.

2. Related Work

Bearing-Only SLAM has received most attention in the current decade. Therefore many of the related approaches are actually very recent. In [9] Deans proposes a combination of a global optimization BA, (Bundle Adjustment) for feature initialization and Kalman Filter for state estimation. In this method, due to the limitation on the baseline on which features can be initialized and depending on the camera motion and the landmark location, some features cannot be initialized. Strelow proposes in [10] a similar method but mixing camera and inertial sensors measurements in an Iterated Extended Kalman Filter (IEKF). In [11] Bailey proposes a variant of constrained initialization for bearing-only SLAM, where past vehicle pose estimates are retained in the SLAM state so that feature initialization can be deferred until their estimates become well-conditioned. In that sense the past poses of the robot are stacked in the map, together with associated measures, until base-line is sufficient to permit a Gaussian initialization. The criteria used for determining whether the estimation is well-conditioned (Gaussian) is the Kullback distance. The complexity of the proposed sampling method to evaluate this distance is very high.

Also there are some works that use other estimation techniques (apart to the EKF) in Bearing-Only SLAM like the Particle Filters (PF) based methods. In a series of papers, Kwok uses Particle Filters: in [12] variations to standard PF are proposed to remedy the sample impoverishment problem in bearing-only SLAM. In [13] initial state of features is approximated using a sum of Gaussians, which defines a set of hypothesis for the position of the landmark, and includes all the features inside the map from the beginning. On successive observations, sequential ratio test (SRT) based on likelihoods is used to prune bad hypothesis. The way these hypotheses are initialized is not detailed, and convergence and consistency issues are not discussed. In [14] Kwok extends the algorithm using a Gaussian Sum Filter, with an approach similar to the proposed in [15] for bearing-only tracking. This method is perhaps the first undelayed feature initialization method. The main drawback of this approach is that the number of required filters can grow exponentially, and therefore computational load grows exponentially with the number of landmarks.

Some of the most notably advances on Bearing-Only SLAM have been presented by Davison [4,16], who shows the feasibility of real-time SLAM with a single camera, using the well-established EKF estimation framework. In this work a Bayesian partial-initialization scheme for incorporating new landmarks are proposed where a separate Particle Filter is used for estimating the feature depth which is not correlated with the rest of the map. In that sense it maintains a set of depth hypotheses uniformly distributed along the viewing ray of a new landmark, with a particle filter in one dimension. Each new observation is used to update the distribution of possible depths, until the variance range is small enough to consider a Gaussian estimation, in this point the estimation is added to the map as a three-dimensional entity. Until this initialization occurs, the ray estimation is maintained in the system's single Extended Kalman Filter. A drawback of this approach is that the initial distribution of particles has to cover all possible depth values for a landmark, this fact makes it difficult to use when the number of detected features is large or when there are far features in the scene. As a result, its application in large environments is not straightforward, as it would require a huge number of particles.

Jensfelt in [17] presents a method where the idea is to let the SLAM estimation lag behind N frames and using these N frames to determine which points are good landmarks and find an estimate of

their 3D location. Mainly the focus in this work is on the management of the features to achieve real-time performance in extraction, matching and loop detection.

In [18] Sola presents a method based on Federate Kalman Filtering technique. With initial Probability Distribution Function (PDF) for the features, a geometric sum of Gaussians is defined. The method is an approximation of the Gaussian Sum Filter (GSF), which was used in [14], that permits undelayed initialization with simply an additive growth of the problem size. A drawback of this approach is that it does not cope with features at very large depths. In [19] Sola presents a similar method to the presented one in [18] but features are initialized with a delayed method.

In [20] a FastSLAM [21] based approach is proposed by Eade. Here the pose of the robot is represented by particles and a set of Kalman Filters refines the estimation of the features. When the inverse depth collapses, the feature is converted to a fully initialized standard Euclidean representation. This approach for features initialization seems appropriate within a FastSLAM implementation, but it lacks for a more general framework.

Table 1. Summary of methods.

Approach	Delayed / Undelayed	Initial representation	Estimation
Deans [9]	Delayed	Bundle adjustment	EKF
Strelow [10]	Delayed	Triangulation	IEKF
Bailey [11]	Delayed	Triangulation	EKF
Davison [16]	Delayed	Multi-Hypotheses	EKF
Kwok (a) [13]	Delayed	Multi-Hypotheses	PF
Kwok (b) [14]	Undelayed	Multi-Hypotheses	PF
Sola [18]	Undelayed	Multi-Hypotheses	EKF
Lemaire [19]	Delayed	Multi-Hypotheses	EKF
Jensfelt [17]	Delayed	Triangulation	EKF
Eade [20]	Delayed	Single Hypotheses	FastSLAM
Montiel [22]	Undelayed	Single Hypotheses	EKF
Munguía[23]	Delayed	Triang.- S. Hypotheses	EKF
This work	Delayed-Undelayed	KF-Triang.- S.	EKF

Montiel *et al.* in [22] presented a method, where the transition from partially to fully initialized features does not need to be explicitly tackled, making it suitable for direct use in EKF framework for sparse mapping. In this approach the features are initialized in the first frame observed with an initial fixed inverse depth and uncertainty, determined heuristically to cover range from nearby to infinity, therefore distant points can be coded. Due to the clarity and scalability, this approach is a good option to be implemented.

On the other hand, experiments show that initial fixed parameters can affect the robustness of the method, especially when an initial metric reference is used in order to recover/set the scale of the map. This fact motivated the authors to develop in [23] a delayed version of the above method. In this case, initial depth and uncertainty of each feature are dynamically estimated prior to add this new landmark

in the stochastic map. The works [22] and [23] are analyzed in the next sections. Table 1 shows a summary of the above methods.

3. Problem Statement

3.1. Sensor motion model

Let us consider a bearing sensor, with a limited field of view, moving freely in 2DOF. The sensor state $\hat{x}_v$ is defined by:

$$\hat{x}_v = [x_v, y_v, \theta_v, v_x, v_y, v_\theta]^T \quad (1)$$

where $[x_v, y_v, \theta_v]$ represents the center position and orientation of the sensor and $[v_x, v_y, v_\theta]$ denoting linear and angular velocity.

At every step it is assumed an unknown linear and angular acceleration with zero mean and known covariance Gaussian processes, a^W and α^W , producing an impulse of linear and angular velocity:

$$n = \begin{bmatrix} V_x^W \\ V_y^W \\ V_\theta^W \end{bmatrix} = \begin{bmatrix} a_x^W \Delta t \\ a_y^W \Delta t \\ a_\theta^W \Delta t \end{bmatrix} \quad (2)$$

The sensor motion prediction model is:

$$f_v = \begin{bmatrix} x_{v(k+1)} \\ y_{v(k+1)} \\ \theta_{v(k+1)} \\ v_{x(k+1)} \\ v_{y(k+1)} \\ v_{\theta(k+1)} \end{bmatrix} = \begin{bmatrix} x_{v(k)} + (v_{x(k)} + V_x^W) \Delta t \\ y_{v(k)} + (v_{y(k)} + V_y^W) \Delta t \\ \theta_{v(k)} + (v_{\theta(k)} + V_\theta^W) \Delta t \\ v_{x(k)} + V_x^W \\ v_{y(k)} + V_y^W \\ v_{\theta(k)} + V_\theta^W \end{bmatrix} \quad (3)$$

An Extended Kalman Filter propagates the sensor pose and velocity estimates, as well as feature estimates.

3.2. Features definition and measurement model

The complete state $\hat{x}$ that includes the features $\hat{y}$ is made of:

$$\hat{x} = [\hat{x}_v, \hat{y}_1, \hat{y}_2, \dots, \hat{y}_n]^T \quad (4)$$

where a feature $\hat{y}$ represents a feature i defined by the 4-dimension state vector:

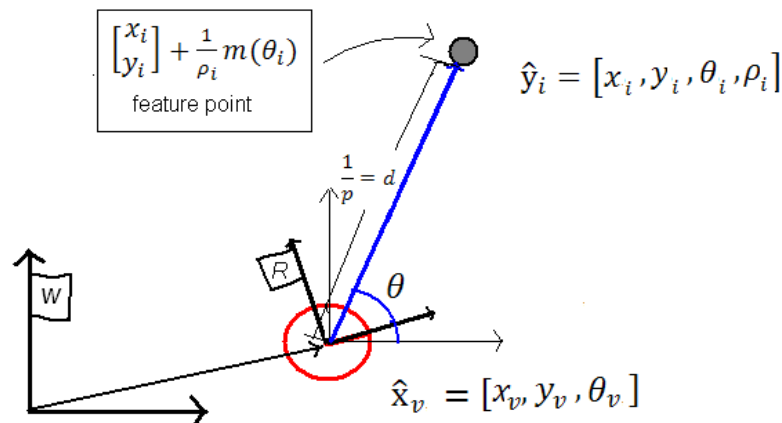
$$\hat{y}_i = [x_i, y_i, \theta_i, \rho_i]^T \quad (5)$$

which models a 2-D point located at:

$$\begin{bmatrix} x_i \\ y_i \end{bmatrix} + \frac{1}{\rho_i} m(\theta_i) \quad (6)$$

where x_i, y_i is the sensor center coordinates when the feature was first observed; and θ_i represents the azimuth (respect to the world reference W) for the directional unitary vector $m(\theta_i)$. The point depth d_i along the ray is coded by its inverse $\rho_i = 1/d_i$ (Figure 1).

Figure 1. Inverse depth parameterization.



The use of an inverse depth parameterization for bearing-only SLAM can improve the linearity of the measurement equation even for small changes in the sensor position (corresponding to small changes in the parallax angle), this fact allows a Gaussian distribution to cover uncertainty in depth which spans a depth range from nearby to infinity. It is well known the relevance of a good uncertainty Gaussian representation in a scheme based in EKF [24].

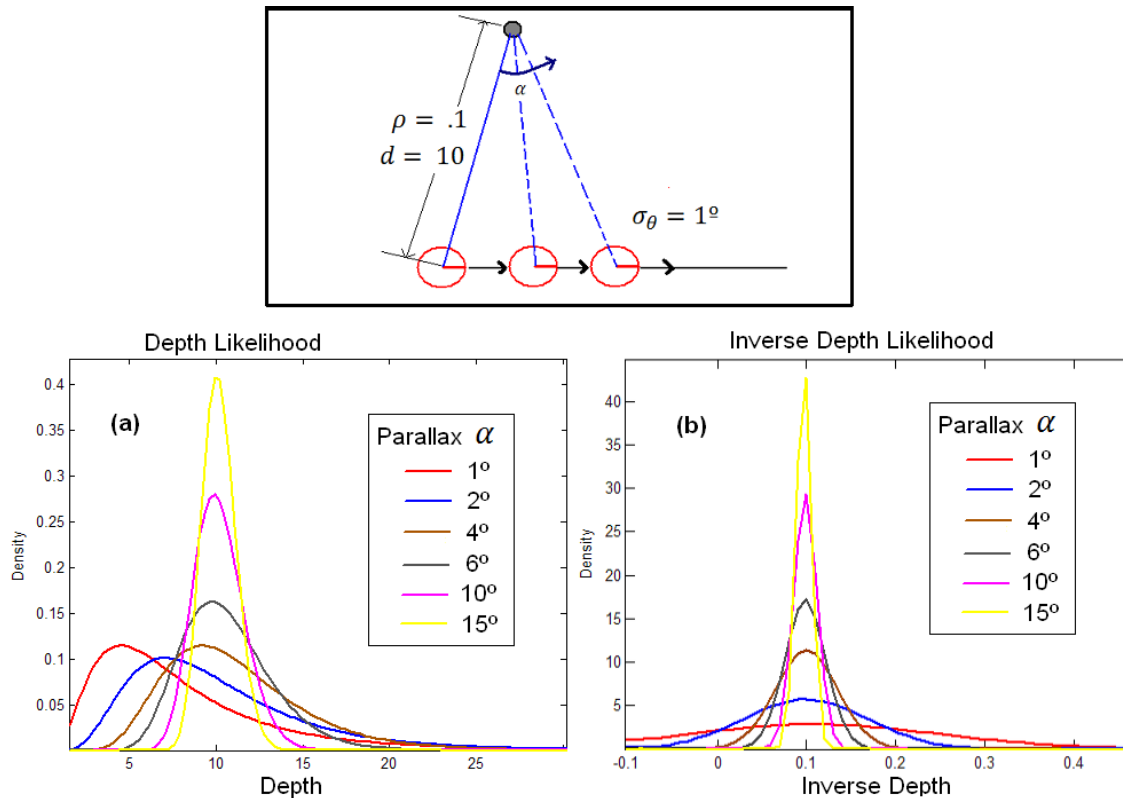
Figure 2 shows a simulation for a point reconstruction from noisy bearing measurements at different parallax (upper plot), using both, the Euclidean and the Inverse Depth parameterization. The location of the vehicle is known. A Gaussian error $\sigma_\theta = 1^\circ$ (degrees) is introduced in bearings. Lower plots show the evolution of the likelihood for depth and inverse depth as the parallax in the observation grows: in (a), the estimates of depth likelihood converge to a Gaussian-like shape, but the initial estimates are highly non-Gaussian, with heavy tails. In contrast, likelihoods of inverse depth (b) (abscissa in inverse meters) are nearly Gaussian, even for low parallax. Therefore it can be clearly appreciated how the uncertainty can be represented by a Gaussian using the inverse depth parameterization over whole parallax range, whereas the Euclidean representation converges to a Gaussian-like shape only to the final estimates. For the Euclidean representation, the parallax needed for the likelihood converges to a Gaussian-like shape, depending on the sensor noise, and thus the noisier the sensor the more parallax is needed for convergence.

The different locations of the sensor, along with the location of the already mapped features, are used to predict the feature angle h_i (angle describing the direction of the feature in the sensor coordinate frame). The measurement model is defined by:

$$h_i = \text{atan2}\left(\frac{1}{\rho_i} \sin \theta_i + y_i - y_v, \frac{1}{\rho_i} \cos \theta_i + x_i - x_v\right) \quad (7)$$

atan2 is a two-argument function that computes the arctangent of y/x given y and x , within a range of $[-\pi, \pi]$. At this stage it is assumed that the bearing sensor is capable of tracking and discriminating between the landmarks, in other words, the data association problem is obviated.

Figure 2. Simulation of a point reconstruction from observations with different parallax.



In implementation using real data, features search could be constrained to regions around the predicted h_i . These regions are defined by the innovation covariance matrix $S_i = H_i P_{k+1} H_i' + R$ where H_i is the Jacobian of the sensor model with respect to the state, P_{k+1} is the prior state covariance, and measurements z are assumed corrupted by zero mean Gaussian noise with covariance R .

As it was stated before, depth information cannot be obtained in a single measurement when bearing sensors are used. To infer the depth of a feature, the sensor must observe it repeatedly as the sensor freely moves through its environment, estimating the angle from the feature to its center. The difference between angle measurements is the feature parallax. Actually, parallax is the key that allows to estimating features depth. In the case of indoor sequences, centimeters are enough to produce parallax, on the other hand, the more distant the feature, the more the sensor has to travel to produce parallax. Therefore, in order to incorporate new features to the map, special techniques for features system-initialization are needed in order to enable the use of bearing sensors in SLAM systems.

Let us consider two methods, which represent the main approaches (undelayed and delayed) for addressing the initialization problem.

3.3. ID-Undelayed initialization

For the Inverse depth (ID) undelayed method presented in [22], transition from partially to fully initialized features do not need to be explicitly tackled; this means that the feature is added to the map in its final representation since the first frame was observed. The initialization includes both the feature state initial values and the covariance assignment. The initial uncertainty region covers a huge range depth $[d_{min}, \infty]$ as Gaussian because the low linearization errors, due to the inverse depth parameterization. Once initialized, the feature is processed with the standard EKF prediction-update loop.

Using the inverse depth parameterization, while the feature is observed at low parallax, the feature will be used mainly to determine the sensor orientation but the feature depth will be kept quite uncertain; if the sensor translation is able to produce a parallax big enough then the feature depth estimation will be improved.

For the ID-Undelayed method (Figure 1), a new feature $\hat{y}_{new}$ (Equation 5) is initialized, when is detected the first time k , as follows:

$$\begin{bmatrix} x_i \\ y_i \\ \theta_i \end{bmatrix} = \begin{bmatrix} x_v \\ y_v \\ \theta_v + z_\theta \end{bmatrix} \quad (8)$$

where x_v, y_v, θ_v are taken directly from the current state $\hat{x}_v$ and z_θ is the initial bearing measurement. The initial value for ρ_i is derived heuristically to cover in its 95% acceptance region, a working space from infinity to a predefined close distance d_{min} expressed as inverse depth:

$$\left[\frac{1}{d_{min}}, 0 \right] \text{ so: } \rho_i = \frac{\rho_{min}}{2} \quad \sigma_\rho = \frac{\rho_{min}}{4} \quad \rho_{min} = \frac{1}{d_{min}} \quad (9)$$

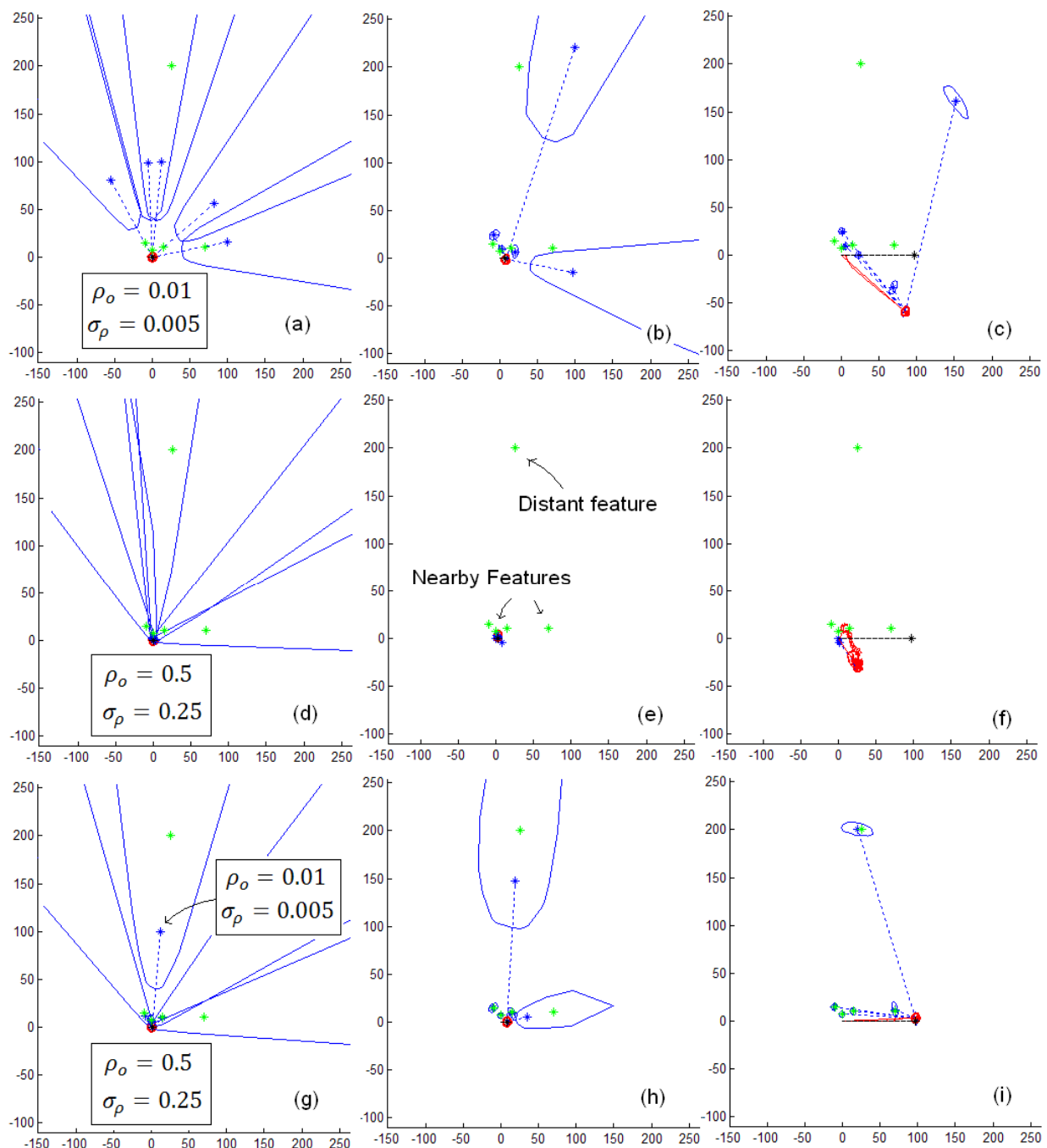
In [22] the parameters are set as $d_{min} = 1$, $\rho_i = 0.5$, $\sigma_\rho = 0.25$. The new system state $\hat{x}$ is conformed simply adding the new feature $\hat{y}_i$ to the final of the vector state:

$$\hat{x} = \begin{bmatrix} \hat{x}_v \\ \hat{y}_1 \\ \hat{y}_2 \end{bmatrix} \quad \hat{x}_{new} = \begin{bmatrix} \hat{x}_v \\ \hat{y}_1 \\ \hat{y}_2 \\ \hat{y}_i \end{bmatrix} \quad (10)$$

The state covariance after feature initialization is defined by:

$$P_{new} = J \begin{bmatrix} P_{(k)} & 0 & 0 \\ 0 & R_j & 0 \\ 0 & 0 & \sigma_\rho^2 \end{bmatrix} J' \quad (11)$$

being J the Jacobian for the initialization function.

Figure 3. Some issues of the ID-Undelayed initialization.

3.4. ID-Delayed initialization

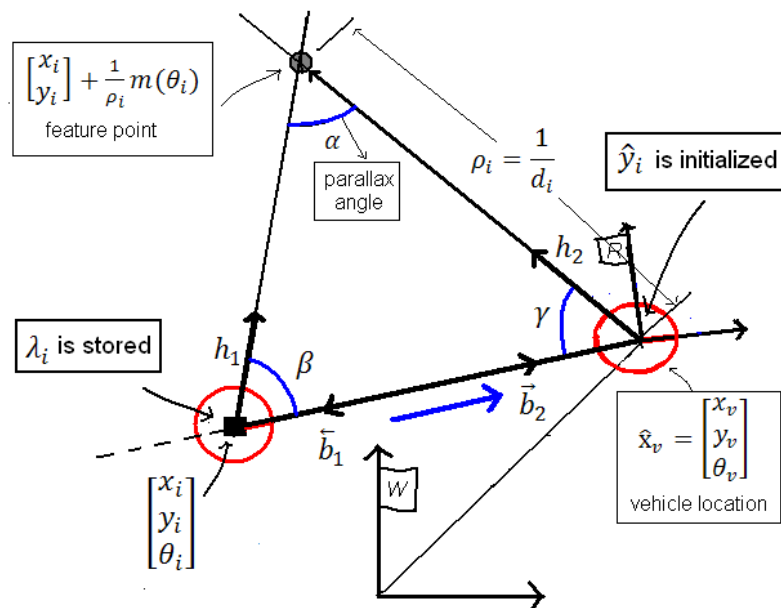
In experiments using the undelayed initialization, it often happens that the inverse depth becomes negative after a Kalman update, due to the observation noise that predominates over the update of the depth, but there are simple solutions to solve this problem. Moreover, when an initial metric reference is used in order to recover/set the scale of the map (very relevant for robotics applications), initial fixed parameters (inverse depth and uncertainty) must be tuned in order to ensure convergence.

Figure 3 illustrates the SLAM process using the ID-Undelayed following a simple straight trajectory for 5 features, (4 of them near to the vehicle and the other more distant). In upper plots (a,b

and c) the initial parameters ($\rho_o = 0.01$, $\sigma_p = 0.005$) are set in order to initialize the features at the middle of the distance between the vehicle and the more distant feature, and therefore far enough respect to the nearby features (plot a). In this case we found in almost every case a huge drift in the estimates (plot c), and it can be appreciated that initializing the features far away from the sensor fails if there are some closest landmarks. If other values ($\rho_o = 0.5$, $\sigma_p = 0.25$) are used (near to the sensor), (central plots d, e and f) then the percentage of convergence is poor (approx 50%) (plot f), due to the influence of the distant feature. In this case if the distant feature is removed from the map (not illustrated here) then a percentage of convergence of 80% is achieved. In the last series (lower plots g, h and i) we combine both initial values: $\rho_o = 0.01$, $\sigma_p = 0.005$ for the distant landmark and $\rho_o = 0.5$, $\sigma_p = 0.25$ for the all the nearby ones. In this case an effectiveness of 90% was achieved for the algorithm.

The issues mentioned above suggest us that the initial inverse depth and their associated initial uncertainty of the new features added to the map could be treated before to be added to the system state instead of use fixed initial depth and uncertainty. In [23] a delayed version of an undelayed method is proposed. In this case, initial depth and uncertainty of each feature are dynamically estimated prior to adding the new landmark in the stochastic map.

Figure 4. ID-Delayed parameterization.



For the ID-Delayed method, a new feature $\hat{y}_{new}$ (Equation 5) is initialized as follows:

When a feature is detected the first time k , some part of the current state $\hat{x}$ and covariance matrix P together with the sensor measurement are stored, this data λ (called *candidate points*) is composed by:

$$\lambda_i = [x_1, y_1, \theta_1, z_1, \sigma_1^x, \sigma_1^y, \sigma_1^\theta] \quad (12)$$

The values x_1 , y_1 and θ_1 represent the current robot position; σ_1^x , σ_1^y and σ_1^θ represent their associated variances taken from the state covariance matrix P_k and z_1 is the first bearing measurement to the landmark. In subsequent instants k , the feature is tracked until a minimum parallax threshold α_{min}

is reached. Figure 4 shows that a few degrees of parallax are enough to reduce the uncertainty in the estimation.

The parallax α is estimated using:

- (i) The base-line b .
- (ii) λ_i , using its associated data $(x_1, y_1, \theta_1, z_1, \sigma^x_1, \sigma^y_1, \sigma^\theta_1)$.
- (iii) The current state $(x_v, y_v, \theta_v, z, \sigma^x_v, \sigma^y_v, \sigma^\theta_v)$.

For each candidate point λ_i , every time that a new bearing measurement z is available, the parallax angle α can be estimated as (Figure 4):

$$\alpha = \pi - (\beta + \gamma) \quad (13)$$

The angle β is determined by the directional unitary vector h_1 and the vector b_1 defines the base-line b in the direction of the sensor trajectory.

The angle γ is determined in a similar way as β but using the directional unitary vector h_2 and the vector b_2 defining the base line in the opposite direction of the sensor trajectory by:

$$\beta = \cos^{-1} \left(\frac{h_1 \cdot b_1}{\|h_1\| \|b_1\|} \right) \quad \gamma = \cos^{-1} \left(\frac{h_2 \cdot b_2}{\|h_2\| \|b_2\|} \right) \quad (14)$$

where $(h_1 \cdot b_1)$ is the dot product between h_1 and b_1 . The directional vector h_1 , expressed in the absolute frame W, points from the sensor location to the direction when the landmark was observed for the first time, and is computed using the data stored in λ_i denoting the bearing z_i . The directional vector h_2 expressed in the absolute frame W is computed in a similar way as h_1 but using the current sensor position $\hat{x}_v$ and the current measurement z_i .

$$h_1 = \begin{bmatrix} \cos(\theta_1 + z_1) \\ \sin(\theta_1 + z_1) \end{bmatrix} \quad h_2 = \begin{bmatrix} \cos(\theta_v + z_i) \\ \sin(\theta_v + z_i) \end{bmatrix} \quad (15)$$

b_1 is the vector representing the robot base-line between the robot center position x_1, y_1 stored in λ_i where the point was first detected and the current sensor center (x_v, y_v) . b_2 is equal to b_1 but pointing to the opposite direction. The base-line b is the module of b_2 or b_1 :

$$b = \|b_1\| = \|b_2\| \quad b_1 = [(x_v - x_1), (y_v - y_1)] \quad b_2 = [(x_1 - x_v), (y_1 - y_v)] \quad (16)$$

If $\alpha > \alpha_{\min}$ then λ_i is initialized as a new feature $\hat{y}_i$. The threshold $\alpha_{\min}$ can be established depending on the accuracy of the bearing sensor. Depth uncertainty is reasonably well minimized when $\alpha = 10^\circ$.

For a new feature $\hat{y}_i$, values of x_i, y_i, θ_i are defined in the same way as Equation 8. For the delayed approach the dynamical estimation of ρ_i is derived from:

$$\rho_i = \frac{\sin \alpha}{b \sin \beta} \quad (17)$$

The variance σ_ρ for the inverse depth ρ is calculated now from the initialization process, instead of a variance predefined heuristically as it was made in the undelayed method, therefore the covariance for

the new feature $\hat{y}_{\text{new}}$ is derived from the error diagonal covariance matrix R_i measurement and the state covariance matrix P .

$$R_i = \text{diag}[\sigma_1^x, \sigma_1^y, \sigma_1^\theta, \sigma_z^2, \sigma_z^2] \quad (18)$$

For reasons of simplicity R_i is defined as a diagonal matrix (cross-covariances are not taken into account) and is now conformed by the error variance of the standard deviation of the bearing sensor σ_z (one for each bearing estimation z_1 and z) and the variances stored in λ_i (σ_1^x , σ_1^y and σ_1^θ). Note that the value of σ_z is constant and is not stored previously in Equation 12.

The new state covariance matrix, after initialization, is:

$$P_{\text{new}} = J \begin{bmatrix} P_k & 0 \\ 0 & R_j \end{bmatrix} J' \quad (19)$$

Note that unlike the ID-Undelayed method there is not an implicit initial uncertainty in depth σ_p (Equation 11). In the ID-Delayed method the complete covariance for the new feature is fully estimated by the initialization process.

3.5. Undelayed vs. Delayed

In an Undelayed approach, when a feature is added to the map the first time that it has been observed, its depth is modeled with a huge uncertainty. In that sense, this new feature does not provide any depth information. However, at this stage the benefit of the Undelayed approach is that features provide information about the sensor orientation from the beginning.

On the other hand, it can be useful to wait until the sensor movement produces some degrees of parallax, (gathering depth information) in order to improve robustness, especially when an initial metric reference is used for recovering scale. Moreover, when cameras are used in real cluttered environments, the delay can be used for efficiently reject weak features, thus initializing only the best features as new landmarks to the map.

4. Concurrent Initialization

This section presents a novel and robust method, called *Concurrent Initialization*, for initializing new features in bearing-sensor-based SLAM systems. The method takes advantage of both, undelayed and delayed approaches. When a feature is detected for the first time, it is immediately initialized in the map (undelayed) as a directional vector which contributes since the beginning to the estimation of the sensor orientation. After that, while the sensor moves freely through its environment, the incoming measurements are incorporated via an uncorrelated linear Kalman filter in order to estimate the depth of the feature. If the sensor movement produces enough parallax, then the feature will be updated with the estimated depth, and thus also contributing to the estimation of the sensor location. Very far features will not produce parallax, and will remain in the form of a directional vector in the map, but contributing to the estimation of sensor orientation. Figure 5 illustrates the concurrent initialization process.

4.1. Undelayed stage

When a feature is detected for the first time k , it is initialized immediately in the map as a new landmark $\hat{y}_{L(i)}$ which is composed by the 3-dimension state vector:

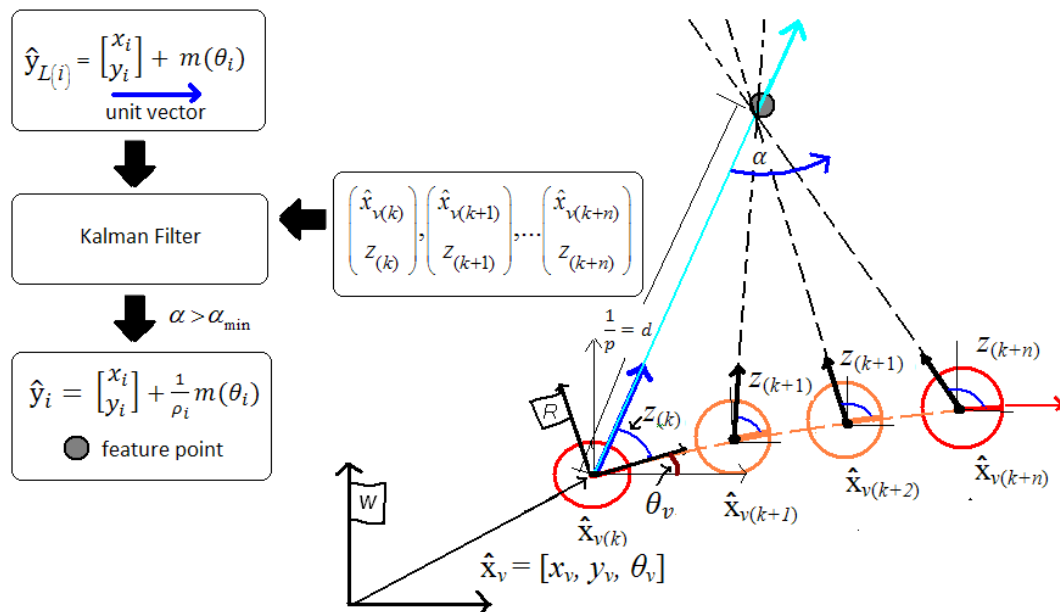
$$\hat{y}_{L(i)} = [x_i, y_i, \theta_i]^T \quad (20)$$

where

$$\begin{bmatrix} x_i \\ y_i \\ \theta_i \end{bmatrix} = \begin{bmatrix} x_v \\ y_v \\ \theta_v + z_\theta \end{bmatrix} \quad (21)$$

x_v, y_v, θ_v are taken directly from the current state $\hat{x}_v$ and z_θ is the initial bearing measurement. $\hat{y}_{L(i)}$ defines a directional vector, expressed in the absolute frame W , which represents the direction of the landmark from the sensor, when it was observed for the first time. The covariance matrix P is updated in the same manner as Equation 19 but using the proper Jacobian J .

Figure 5. Concurrent initialization process.



Parallel to the system state (represented by the state vector $\hat{x}$), a state vector $\hat{x}_{can}$ is used for estimating (via an extra linear Kalman Filter) the feature depth of each landmark $\hat{y}_L$. The state $\hat{x}_{can}$ is not directly correlated with the map.

Every time a new feature $\hat{y}_{L(i)}$ is initialized in $\hat{x}$, the state $\hat{x}_{can}$ is augmented as:

$$\hat{\mathbf{x}}_{\text{can}} = \begin{bmatrix} \lambda_1 \\ \lambda_2 \end{bmatrix} \quad \hat{\mathbf{x}}_{\text{can_new}} = \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_1 \end{bmatrix} \quad (22)$$

where λ_i is a 3-dimension vector composed by:

$$\lambda_i = [\alpha_i, \Delta\alpha_i, \rho_i]' \quad (23)$$

For each λ_i , α_i is the estimated parallax, $\Delta\alpha_i$ is the rate of change in parallax and ρ_i is the estimated inverse depth.

The covariance matrix of $\hat{\mathbf{x}}_{\text{can}}$, $\mathbf{P}_{\text{can}}$, is augmented simply by:

$$\mathbf{P}_{\text{can_new}} = \begin{bmatrix} \mathbf{P}_{\text{can}} & 0 \\ 0 & \mathbf{R}_c \end{bmatrix} \quad (24)$$

The three initial values of λ_i are set to zero, and the initial values of $\mathbf{R}_c$ have been heuristically determined as: $\mathbf{R}_c = \text{diag}(0.01, 0.01, 1)$.

4.2. Delayed stage

While the sensor moves through its environment, it can observe repeatedly a landmark $\hat{\mathbf{y}}_{L(i)}$, at each iteration generating a new angle measurement z . All these new measurements are successively added to the linear Kalman Filter (responsible for estimating $\hat{\mathbf{x}}_{\text{can}}$) in order to infer the landmark depth. For each new measurement z_i of a feature $\hat{\mathbf{y}}_{L(i)}$ an iteration of the filter is executed.

The state transition model for each λ_i is:

$$\hat{\mathbf{x}}_{\text{can_i}(k+1)} = \begin{bmatrix} \alpha_{i(k+1)} \\ \Delta\alpha_{i(k+1)} \\ \rho_{i(k+1)} \end{bmatrix} = \begin{bmatrix} \alpha_{i(k)} + \Delta\alpha_{i(k)} \\ \Delta\alpha_{i(k)} \\ \rho_{i(k)} \end{bmatrix} \quad (25)$$

A process noise $w_k \sim N(0, Q_k)$ is considered. In experiments: $Q_k = \text{diag}(8e^{-7}, 10e^{-9})$ have been used.

The measurement prediction model is directly obtained from the state. On the other hand, the measurements z_{can} used to update the filter are a function of: (i) the feature $\hat{\mathbf{y}}_{L(i)}$, (ii) the sensor state $\hat{\mathbf{x}}_v$ and (iii) the current measurement z_i .

$$z_{\text{can}} = \begin{bmatrix} z_\alpha \\ z_\rho \end{bmatrix} = f_z(\hat{\mathbf{y}}_{L(i)}, \hat{\mathbf{x}}_v, z_i) \quad (26)$$

z_α and z_ρ are estimated in the same manner as Equation 13 and Equation 17 respectively. Only note that in Equation 16, x_1 and y_1 are taken from $\hat{\mathbf{y}}_{L(i)}$, so $x_1 = x_i$ and $y_1 = y_i$.

The implicit uncertainties in the estimation of the function f_z are used to compose the error measurement covariance matrix $\mathbf{R}_{\text{can}}$:

$$\mathbf{R}_{\text{can}} = \nabla f_z(\mathbf{P}_t) \nabla f_z' \quad (27)$$

where ∇f_z is the Jacobian of f_z with respect to z_{can} . $\mathbf{P}_t$ is formed by:

$$P_t = \begin{bmatrix} P_{\hat{x}_v} & P_{\hat{x}_v \hat{y}_{L(i)}} & 0 \\ P_{\hat{y}_{L(i)} \hat{x}_v} & P_{\hat{y}_{L(i)}} & 0 \\ 0 & 0 & \sigma_z^2 \end{bmatrix} \quad (28)$$

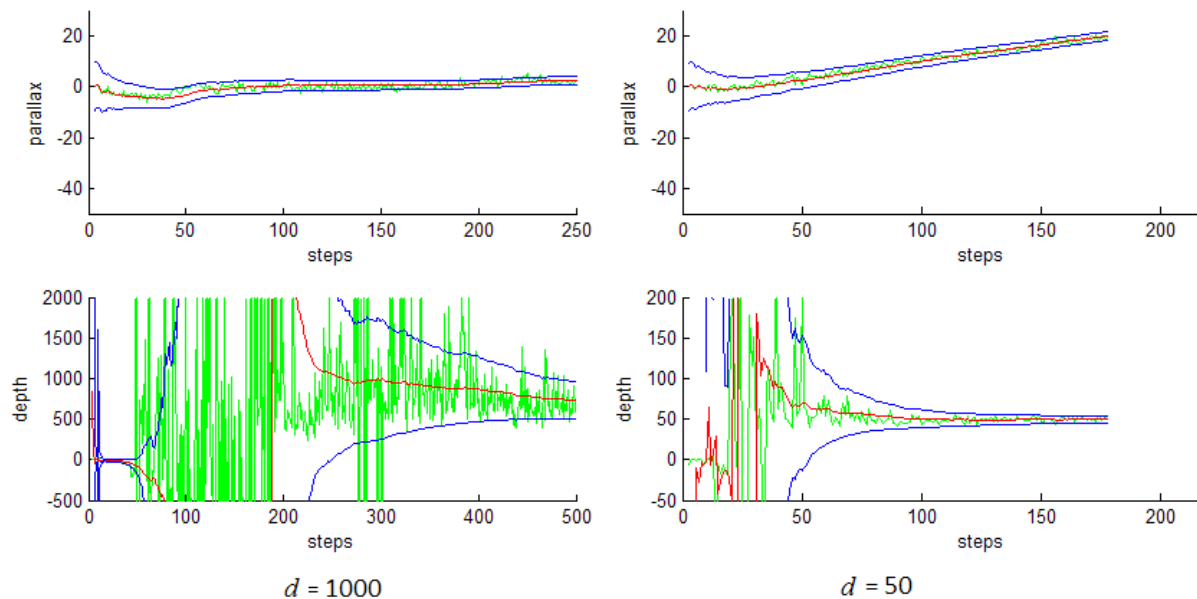
All the components of P_t , except σ_z (the error variance of the bearing sensor) are taken directly from the covariance matrix P of the system state $\hat{x}$. $P_{\hat{x}_v}$ is the submatrix of P corresponding to the covariance of the sensor state $\hat{x}_v$. $P_{\hat{y}_{L(i)}}$ is the covariance of the feature $\hat{y}_{L(i)}$. $P_{\hat{x}_v \hat{y}_{L(i)}}$ and $P_{\hat{y}_{L(i)} \hat{x}_v}$ are the correlations between $\hat{x}_v$ and $\hat{y}_{L(i)}$.

R_{can} is used in the Kalman update equations for estimating the innovation covariance matrix S_i .

4.3. Updating depth

Features expressed in the form of $\hat{y}_{L(i)}$ are very useful to estimate the sensor orientation θ_v . In [25] a visual compass is proposed based in this fact. Besides, the depth of features are needed for estimating the sensor location $[x_v, y_v]$. For near features $\hat{y}_{L(i)}$, a small sensor translation is enough to produce some parallax and thus to infer depth.

Figure 6. Parallax and depth estimations for a distant and a close feature.



The state $\hat{x}_{can}$ encloses the parallax α_i and inverse depth ρ_i estimations for each feature $\hat{y}_{L(i)}$. Figure 6 shows the evolution of parameters α_i (upper plot) and $d = 1/\rho_i$ (lower plot) and its uncertainty for the feature estimated by the linear Kalman filter, (left plot) feature at a distance of $d = 1,000$ units and (right plot) for a distance of $d = 50$ units. The boundary uncertainties at 3σ are indicated in blue color. The filtered values are depicted in red color. Also note in green color the raw measurements z_{can} (taken from Equation 26). In these graphics it can be clearly appreciated how the estimation of depth d is directly influenced by the parallax; for the near feature, about 100 steps are needed to producing

parallax and thus d converges rapidly to its real value. Also note that the uncertainty is rapidly minimized. On the other hand, the distant feature produces small parallax. For a low parallax the sensor noise predominates and therefore produces very fluctuant raw measurements. Even so, after several steps, the filter estimates the depth reasonable well with a high related uncertainty.

A minimum parallax threshold $\alpha_{\min}$ is used for updating a feature $\hat{y}_{L(i)}$ as $\hat{y}_i$. Distant features will not produce parallax and therefore will remain expressed as $\hat{y}_{L(i)}$, but contributing to the estimation of sensor orientation θ_v .

On the other hand, if $\alpha_i > \alpha_{\min}$ then:

$$\hat{y}_{L(i)} = [x_i, y_i, \theta_i]^T \rightarrow \hat{y}_i = [x_i, y_i, \theta_i, \rho_i]^T \quad (29)$$

where ρ_i is taken directly from $\hat{x}_{\text{can}}$. The covariance matrix P is transformed by the corresponding Jacobian:

$$P_{\text{new}} = J \begin{bmatrix} P & 0 \\ 0 & q\sigma_{\rho_i}^2 \end{bmatrix} J' \quad (30)$$

being σ_{ρ_i} the variance of the inverse depth estimation for $\hat{y}_{L(i)}$ and taken from P_{can} . The constant q is used to increase the initial uncertainty of ρ_i in order to improve filter consistency. In experiments q is set to 100.

$$J = \begin{bmatrix} I & 0 & 0 & 0 \\ 0 & \frac{\partial \hat{y}_i}{\partial \hat{y}_{L(i)}} & 0 & 1 \\ 0 & 0 & I & 0 \end{bmatrix} \quad (31)$$

When a feature $\hat{y}_{L(i)}$ is updated as $\hat{y}_i$, then its corresponding values will be removed from the linear Kalman filter responsible for estimating the state $\hat{x}_{\text{can}}$.

4.4. Measurement

At any time, the map can include both kind of features $\hat{y}_{L(i)}$ and $\hat{y}_i$. Thus each kind of feature has its own measurement prediction model:

- For features $\hat{y}_i$ measurement Equation 7 is used. A value of 2 to 6 times the real error of the bearing sensor is considered for the measurement process.
- Features $\hat{y}_{L(i)}$ are supposed to be very far from the sensor and therefore it is assumed that its corresponding bearing measurement z_i will remain almost constant. The measurement prediction model is simply:

$$h_{L(i)} = \theta_i - \theta_v \quad (32)$$

- For near features $\hat{y}_{L(i)}$, the bearing measurement z_i will change rapidly. Due to this fact the standard deviation of the bearing sensor σ_z for features $\hat{y}_{L(i)}$, (in update Kalman equations) is

multiplied by a high value c , ($c = 10e^{10}$ were used in experiments). An interesting issue, to be treated for further work, could be the dynamical estimation of parameter c .

Both kinds of measurement prediction models must be used together, if several measurements z_i are made at the same time for both kinds of features. For example, consider that the bearing sensor takes measurements of the features $\hat{y}_{L(2)}$ and $\hat{y}_3$ simultaneously, then:

$$v = \begin{bmatrix} z_2 \\ z_3 \end{bmatrix} - \begin{bmatrix} h_{L(2)} \\ h_3 \end{bmatrix} \quad R = \begin{bmatrix} c\sigma_z^2 & 0 \\ 0 & \sigma_z^2 \end{bmatrix} \quad (33)$$

$$\nabla H = \begin{bmatrix} \frac{\partial h_{L(2)}}{\partial \hat{x}_v} & 0 & \frac{\partial h_{L(2)}}{\partial \hat{y}_{L(2)}} & 0 & 0 \\ \frac{\partial h_3}{\partial \hat{x}_v} & 0 & 0 & \frac{\partial h_3}{\partial \hat{y}_3} & 0 \end{bmatrix} \quad (34)$$

where v is the innovation, R is the error covariance matrix and ∇H is the Jacobian measurement model.

5. Experiments

Several simulations have been executed in order test the performance of the proposed method in relation to other approaches. Simulations are extremely helpful when different methods are compared among others, because numerous variables (inherent to real environments) are obviated (e.g., data association), or become constants (*i.e.*, synthetic map), therefore the cores of the methods are compared.

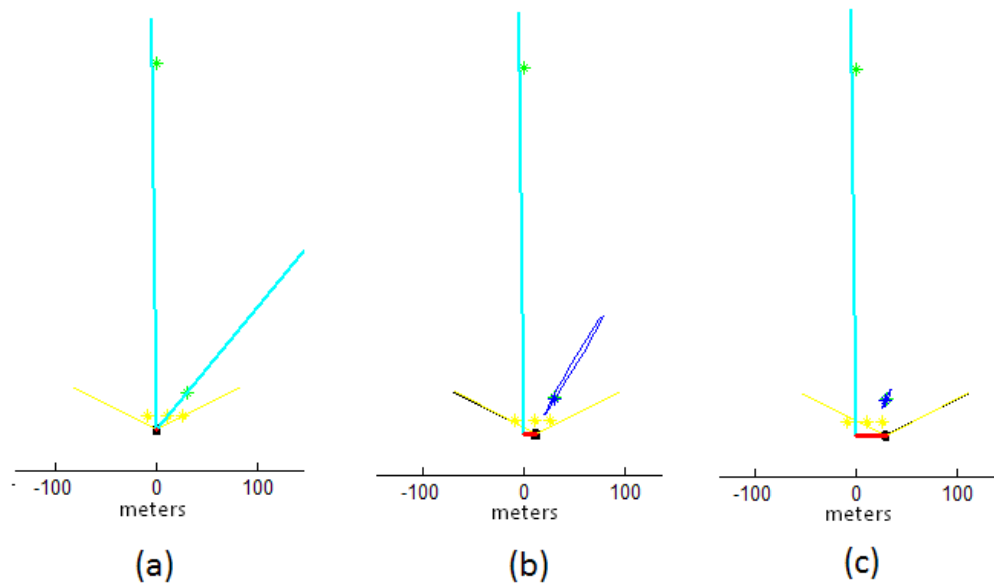
5.1. Initialization process of distant and near features

Figure 7 shows the evolution in the initialization process of two features maps with the concurrent initialization: a distant one (600 units of depth) and a near one (50 units of depth). The only information given a priori to the system was the scale reference (the three points in yellow) which was introduced with an associated uncertainty close to zero in the covariance matrix R . Taking into account that there is not an additional sensory input (e.g., odometry), at every step an unknown linear and angular acceleration is introduced with zero mean and known-covariance Gaussian processes (Section 3.1). In this case, $a_x^w = 4 \text{ m/s}^2$, $a_y^w = 4 \text{ m/s}^2$ and $a_\theta^w = 2 \text{ rad/s}^2$ were used. The only input sensor of the system is a noisy sensor capable of measuring the bearing of features, with a limited field of view of 110° (emulating a 2-DOF camera). A standard deviation of $\sigma = 1^\circ$ is considered for the sensor readings.

At the begin of the sequence (plot *a*) both features has been initialized as a directional vector $\hat{y}_{L(i)}$, defining a ray (in cyan color). Note that three feature points (in yellow color) have been previously added to the map as a priori known landmarks in order to define/set the scale of the world. Around step 100 (plot *b*), the small displacement of the sensor to the right produces enough parallax to estimating the depth of the near feature. Observe that the near feature $\hat{y}_{L(i)}$ has been transformed to a $\hat{y}_i$ feature (blue color). Also note that some uncertainty (especially in depth) remains at the moment of the

transformation. By the last step, at 250 iterations, (plot *c*) the uncertainty in the near feature has been full minimized, on the other hand, the movement of the sensor has not produced enough parallax and therefore, the distant feature remains in the form of $\hat{y}_{L(i)}$ but still contributing to the estimation of the sensor orientation.

Figure 7. Sequence illustrating the concurrent initialization process: (a) beginning of initialization; (b) initialization process at step 100; (c) initialization process at step 250.



5.2. Comparative study

In order to show the performance of the Concurrent method proposed in this article, a comparative study between the ID-Undelayed method [22] and the ID-Delayed [23] method is presented.

Figure 8 illustrates the environment setup used in the study. For all the tests, the bearing sensor is moved over a semi-cycled U-like shape trajectory, since our main goal is to observe the effect of the initialization process of new features in the estimation of both map and sensor location, instead of the closing loop problem. About 100 landmarks (in green) simulate the environment of the sensor. Figure 8 also shows both the map and sensor trajectory estimates after a single run of 2,000 steps of the Concurrent method. The features map and their uncertainties are indicated in blue. Note the evolution of uncertainty in both sensor and features location. Also note the typical (in SLAM systems) drift in both trajectory and map estimations as the sensor moves far away from its initial location.

The *average* NEES (normalised estimation error squared [26]) over *N* Monte Carlo runs of the filter was used in order to evaluate the consistency of the methods, as it is proposed in [27]. The NEES is estimated as follows:

$$\mathcal{E}_k = \left[\mathbf{x}_k - \hat{\mathbf{x}}_{k|k} \right]^T \mathbf{P}_{k|k}^{-1} \left[\mathbf{x}_k - \hat{\mathbf{x}}_{k|k} \right] \quad (35)$$

where $\mathbf{x}_k$ is the true state and the average NEES is computed as:

$$\bar{\varepsilon}_k = \frac{1}{N} \sum_{i=1}^N \varepsilon_{k(i)} \quad (36)$$

Four different tests were realized to comparing the methods under diverse conditions. Table 2 shows the values for the linear and angular acceleration (a_x^w , a_y^w and a_θ^w in 4m/s^2) and the time between “frames” (Δt in seconds) used for each test. In this case, a higher Δt implies bigger displacements of the robot from frame to frame and therefore more linearization errors due to large changes in parallax.

Table 2. Setup of the tests.

Test	a_x^w	a_y^w	a_θ^w	Δt
a	4	4	2	1/30
b	4	4	2	1/120
c	6	6	3	1/30
d	6	6	3	1/120

In experiments $\rho_{ini} = 0.05$ and $\sigma_p = 0.025$ were used for the ID-Undelayed method and $\alpha_{min} = 10^\circ$ were used for both ID-Delayed and Concurrent methods. The NEES was estimated over the 3-dimensional robot pose. The average NEES for each method was estimated using $N=20$ Monte Carlo runs. In simulations the sensor was moved 3 meters every $\Delta t = 1$ second for the straight sections and 4.5 meters every $\Delta t = 1$ seconds for the curve sections of the trajectory.

MATLAB code was run using a 1.73 GHz Pentium M laptop. Table 3 shows the execution time of the three methods for conditions $\Delta t = 1/30$ and $\Delta t = 1/120$. As it would be expected, for $\Delta t = 1/120$ the execution time was around four times longer than $\Delta t = 1/30$, for the same trajectory. Execution time of the Concurrent method was estimated using a diagnostic version of the algorithm, marked with an asterisk (*) in Table 3. This version uses a $5n$ Kalman filter (instead of $3n$) which estimates two extra diagnostic parameters. This means that the real execution time of the Concurrent method should be somewhat faster.

Table 3. Execution time & Failed attempts.

Method	$\Delta t=1/30$	$\Delta t=1/120$	a	b	c	d
ID-Undelayed	76s	302s	2	1	0	2
ID-Delayed	60s	235s	9	2	11	4
Concurrent	94s*	376s*	0	0	0	0

For some runs of the algorithms, filter divergence could occur. Table 3 also shows the failed attempts, this means the number of times that filter divergence appeared before a method reach $N = 20$ positives runs (with convergence), for each test. The average NEES was only estimated using positive runs.

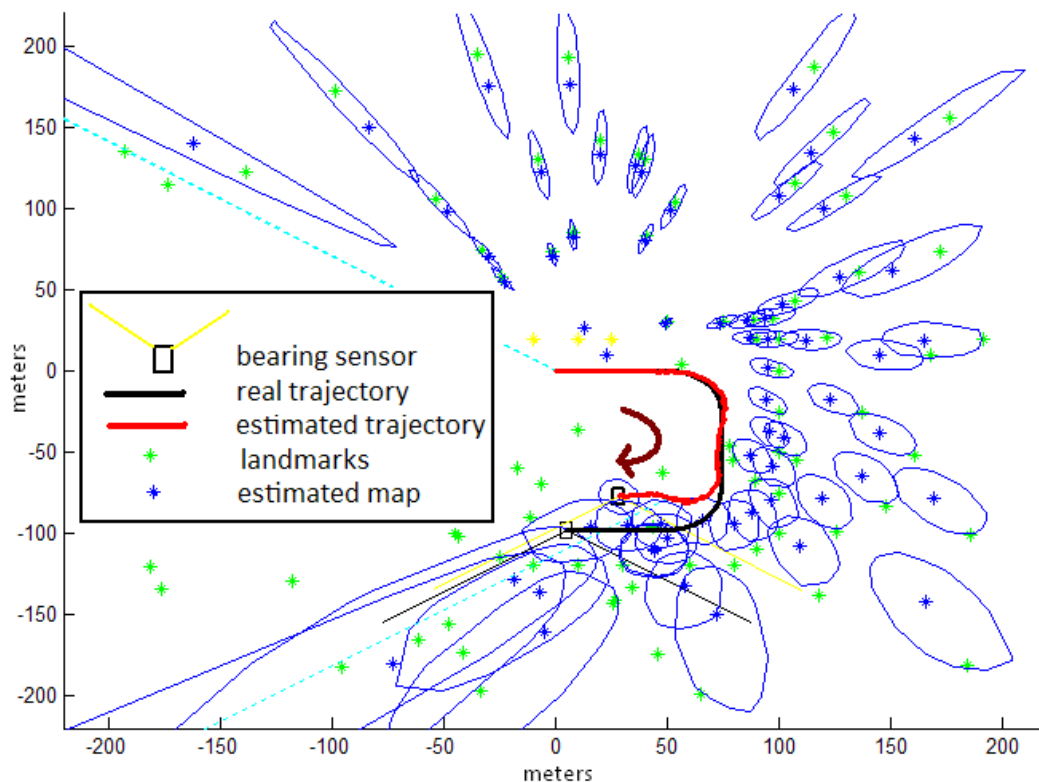
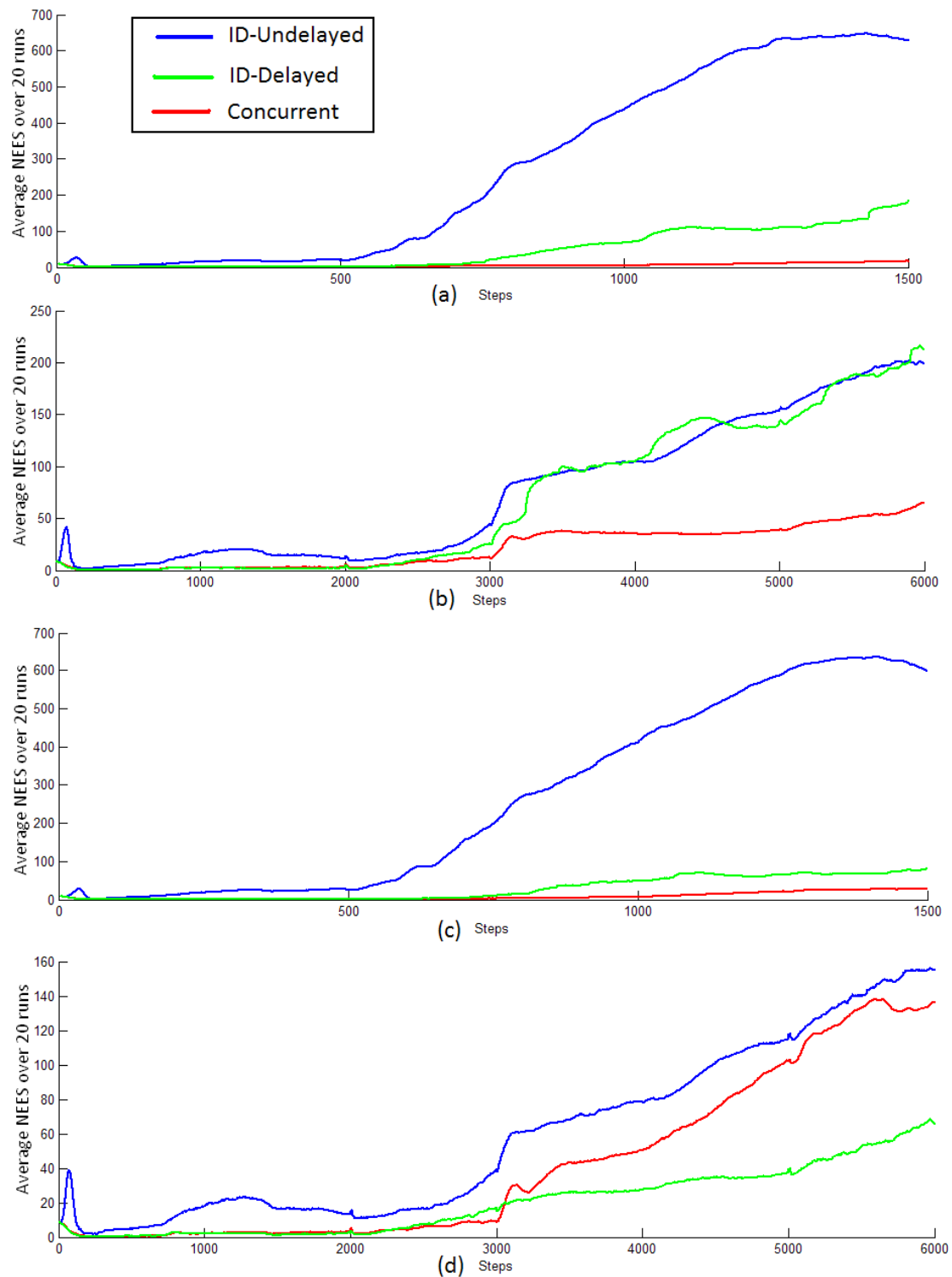
Figure 8. SLAM simulation using the concurrent initialization method.

Figure 9 shows the evolution of the average NEES ($\bar{\varepsilon}_k$) for tests (a), (b), (c), (d) for each method. Note that test (b) and (d) need four times more steps than test (a) and (c) in order to complete the same trajectory due to their particular value of Δt .

As it would be expected, the whole methods become optimistic after a certain time [27]. Nevertheless, it can be observed for tests (a), (b) and (c) that $\bar{\varepsilon}_k$ achieved for Concurrent method (red) tends to be lower than $\bar{\varepsilon}_k$ achieved for both ID-Undelayed (blue) and ID-Delayed (green) methods. This behavior is more evident in test (a) and (c), where lower noise is injected into the sensor motion model. The difference between (a) and (c) is the time between “frames” (Δt) used for each test. The above results suggest, that if adequate noise is injected, then Concurrent method seems to be less sensitive to large jumps in parallax (and linearization errors), and therefore could be suitable for application where a high “frame-rate” is not available. Regarding to the ID-Undelayed method, it can be observed the effect of the parameter Δt over the magnitude of $\bar{\varepsilon}_k$, although its form is somewhat similar for all tests. In test (a) and (c) where Δt is higher (1/30), $\bar{\varepsilon}_k$ reaches around 600 units, while in test (b) and (d) where Δt is lower (1/120) $\bar{\varepsilon}_k$ reaches around 200 units. Theoretically the $\bar{\varepsilon}_k$ for the ID-Undelayed method would be more favorable as $\Delta t \rightarrow 0$. At this point, it is important to remember that implementation of Concurrent method implies the estimation of an extra linear Kalman filter of dimension $3n$, where n is the number of features $\hat{y}_{L(i)}$ in the state $\hat{x}$. On the other hand, the Concurrent method is less sensitive to the parameter Δt . At least for this experimental setup, the average NEES for the Concurrent method, when $\Delta t = 1/30$ (94 s of execution time), is even lower than the $\bar{\varepsilon}_k$ for the ID-Undelayed, when

$\Delta t = 1/120$ (302 s of execution time). The ID-Delayed method represents the most efficient alternative in computational cost with medium performance in average NEES terms. However the ID-Delayed method shows to be the least robust method in terms of convergence (Table 3). The problem of convergence for the ID-Delayed method is notorious in tests (a) and (c) where Δt is higher (1/30), these results indicate a frame-rate dependency of the method. On the other hand, for the realized tests, the Concurrent method shows the best results in terms of convergence.

Figure 9. Comparison of the average NEES for ID-Undelayed, ID-Delayed and Concurrent methods.



6. Conclusions

This work proposes a novel and robust approach for initializing new features in SLAM systems based in bearing sensors. First, an overview of the problem is given and the most relevant related work is presented. Most of the methods presented in the literature can be classified into two categories: Undelayed and Delayed Methods. In that sense, an analysis of two representative methods of the aforementioned taxonomy is also presented. Undelayed methods provide information of orientation

since a feature is first detected, on the other hand, Delayed methods await until some depth information is gathered, improving convergence.

The proposed approach in this article, the concurrent initialization method, takes the best of both, Undelayed and Delayed approaches. The key is to use two kinds of feature representations concurrently for both undelayed and delayed stages of the estimation. The simulations results, based in the average NEES test, showed that the Concurrent method can maintain the filter consistency satisfactorily. Moreover, observing the percentage of convergence, the Concurrent method appears also to be robust.

It is important to mention that the complexity of the Concurrent method is higher than other methods (*i.e.*, ID-Undelayed method) and the additional Kalman filter certainly implies an increase of computational requirements per frame. On the other hand, the Concurrent method appears to be less sensitive to the linearization errors induced for large jumps in parallax (much time between frames). In that sense the Concurrent method could be even more efficient in computational terms than other methods because it seems to work properly at low frame rate. This attribute also makes it suitable for applications where a high frame rate is not available for different reasons. The concurrent initialization method could be a robust alternative to bearing sensor based SLAM systems.

Acknowledgements

This research was conducted in the Automatic Control Department, Universitat Politècnica de Catalunya (UPC) and supported by the Spanish Ministry of Science and Innovation, UbROB Project, CICYT number DPI2007-61452.

References

1. Vázquez-Martín, R.; Núñez, P.; Bandera, A.; Sandoval, F. Curvature-based environment description for robot navigation using laser range sensors. *Sensors* **2009**, *9*, 5894–5918.
2. Durrant-Whyte, H.; Bailey, T. Simultaneous localization and mapping: part I. *IEEE Rob. Autom. Mag.* **2006**, *13*, 99–110.
3. Bailey, T.; Durrant-Whyte, H. Simultaneous localization and mapping: part II. *IEEE Rob. Autom. Mag.* **2006**, *13*, 108–117.
4. Davison, A. Real-time simultaneous localization and mapping with a single camera. In *Proceedings of the ICCV'03*, Nice, France, October 13–16, 2003.
5. Munguía, R.; Grau, A. Single sound source SLAM. In *Lecture Notes in Computer Science*; Springer: Berlin, Germany, 2008; Volume 5197, pp. 70–77.
6. Williams, G.; Klein, G.; Reid, D. Real-time SLAM relocalisation. In *Proceedings of the ICCV'07*, Rio de Janeiro, Brazil, October 14–20, 2007.
7. Chekhlov, D.; Pupilli, M.; Mayol-Cuevas, W.; Calway, A. Real-time and robust monocular SLAM using predictive multi-resolution descriptors. In *Advances in Visual Computing*; Springer: Berlin/Heidelberg, Berlin, Germany, 2006; pp. 276–285.
8. Munguía, R.; Grau, A. Closing loops with a virtual sensor based on monocular SLAM. *IEEE Trans. Instrum. Measure.* **2009**, *58*, 2377–2385.

9. Deans, H.; Martial, M. Experimental comparison of techniques for localization and mapping using a bearing-only sensor. In *Proceedings of International Symposium on Experimental Robotics*, Waikiki, HI, USA, December 11–13, 2000.
10. Strelow, S.; Sanjiv, D. Online motion estimation from image and inertial measurements. In *Proceedings of Workshop on Integration of Vision and Inertial Sensors (INERVIS'03)*, Coimbra, Portugal, June 2003.
11. Bailey, T. Constrained initialisation for Bearing-Only SLAM. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA'03)*, Taipei, Taiwan, September 14–19, 2003.
12. Kwok, N.M.; Dissanayake, G. Bearing-only SLAM in indoor environments. In *Proceedings of Australasian Conference on Robotics and Automation*, Brisbane, Australia, December 1–3, 2003.
13. Kwok, N.M.; Dissanayake, G. An efficient multiple hypotheses filter for bearing-only SLAM. In *Proceedings of International Conference on Intelligent Robots and Systems (IROS'04)*, Sendai, Japan, September 28–October 2, 2004.
14. Kwok, N.M.; Dissanayake, G. Bearing-only SLAM using a SPRT based gaussian sum filter. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA'05)*, Barcelona, Spain, April 18–22, 2005.
15. Peach, N. Bearing-only tracking using a set of range-parametrised extended Kalman filters. *IEE Proc. Control Theory Appl.* **1995**, *142*, 21–80.
16. Davison, A.; Gonzalez, Y.; Kita, N. Real-Time 3D SLAM with wide-angle vision. In *Proceedings of IFAC Symposium on Intelligent Autonomous Vehicles*, Lisbon, Portugal, July 5–7, 2004.
17. Jensfelt, P.; Folkesson, J.; Kragic, D.; Christensen, H. Exploiting distinguishable image features in robotics mapping and localization. In *Proceedings of European Robotics Symposium*, Palermo, Italy, March 16–17, 2006.
18. Sola, J.; Devy, M.; Monin, A.; Lemaire, T. Undelayed initialization in bearing only SLAM. In *Proceedings of IEEE International Conference on Intelligent Robots and Systems*, Edmonton, Canada, August 2–6, 2005.
19. Lemaire, T.; Lacroix, S.; Sola, J. A practical bearing-only SLAM algorithm. In *Proceedings of IEEE International Conference on Intelligent Robots and Systems*, Edmonton, Canada, August 2–6, 2005.
20. Eade, E.; Drummond, T. Scalable monocular SLAM. In *Proceedings of IEEE Computer Vision and Pattern Recognition*, New York, NY, USA, June 17–22, 2006.
21. Montemerlo, M.; Thrun, S.; Koller, D.; Wegbreit, B. FastSLAM: Factored solution to the simultaneous localization and mapping problem. In *Proceedings of National Conference on Artificial Intelligence*, Edmonton, AB, Canada, July 28–29, 2002.
22. Montiel, J.M.M.; Civera, J.; Davison, A. Unified inverse depth parametrization for monocular SLAM. In *Proceedings of Robotics: Science and Systems Conference*, Philadelphia, PA, USA, August 16–19, 2006.
23. Munguía, R.; Grau, A. Delayed inverse depth monocular SLAM. In *Proceedings of the 17th IFAC World Congress*, Coex, Seoul, Korea, July 6–11, 2008.

24. Andrade-Cetto, J.; Sanfeliu, A. *Environment Learning for Indoor Mobile Robots. A Stochastic State Estimation Approach to Simultaneous Localization and Map Building (Springer Tracts in Advanced Robotics)*; Springer: Berlin, Germany, 2006; Volume 23.
25. Montiel, J.M.M.; Davison, A. Visual compass based on SLAM. In *Proceedings of IEEE International Conference on Robotics and Automation*, Orlando, FL, USA, May 15–19, 2006.
26. Bar-Shalom, Y.; Li, X.R.; Kirubarajan, T. *Estimation with Applications to Tracking and Navigation*; John Wiley & Sons: Hoboken, NJ, USA, 2001.
27. Bailey, T.; Nieto, J.; Guivant, J.; Stevens, M.; Nebot, E. Consistency of the ekf-slam algorithm. In *Proceedings of IEEE International Conference on Intelligent Robots and Systems*, Beijing, China, December 4–6, 2006.

© 2010 by the authors; licensee Molecular Diversity Preservation International, Basel, Switzerland. This article is an open-access article distributed under the terms and conditions of the Creative Commons Attribution license (<http://creativecommons.org/licenses/by/3.0/>).