

# Experimental Validation of a PCA-Based Localization System for Mobile Robots in Unstructured Environments

F. Carreira, C. Christo, D. Valério, M. Ramalho, C. Cardeira, J. M. F. Calado, and P. Oliveira

**Abstract**—This paper proposes and experimentally validates an on-board feature based positioning sensor and localization system for mobile robots, rooted on Principal Component Analysis (PCA), to operate in unstructured environments. The positioning system resorts to principal component analysis of images acquired by a video camera installed on-board, looking upwards to the ceiling. No beacons or special features are required to be present in the acquired images. The principal components of the acquired images are compared with previously registered images present in a reduced on-board image database. The position measured is fused with odometry data. A Linear Parameter Varying (LPV) model for the Mobile Robot Kinematics is defined, avoiding local linearizations and the Extended Kalman Filter (EKF) stability limitations. The estimates of position and angular slippage are provided by a linear Kalman filter, with global stable error dynamics. The experimental validation reported in this work focus on the results of a set of exhaustive experiments carried out in a real environment, where the robot travels in a 2D horizontal plane. A small position error estimate was always observed, for arbitrarily long experiments, with global stability, and angular slippage was estimated accurately in real time.

## I. INTRODUCTION

The problem of localization in unstructured environments, without resorting to external sensors and with bounded error estimates, has been a major challenge to the scientific community in the area of mobile robotics; see [9], [4] and the references therein. The inputs to the localization system are the measurements provided by the sensor package installed on-board, like compasses, accelerometers, cameras, time of flight cameras, encoders, ..., and the robot has to autonomously use these on-board sensors to be able to look at the environment and rapidly answer the questions: where am I? what am I facing?

SLAM (Simultaneous Localization And Mapping) is a process by which a mobile robot can build a map of an environment and at the same time to use this map to estimate its localization. In SLAM, both the trajectory of the platform and the localization of all landmarks are estimated online without the need for any a priori knowledge of their localization [9], [28]. However, substantial issues remain to be solved in practice. One of the issues that remain open is that the solutions rely on landmarks or on any other features

C. Christo, D. Valério, M. Ramalho and C. Cardeira are with IDMEC / Instituto Superior Técnico, Technical University of Lisbon, Av. Rovisco Pais 1, 1049-001 Lisboa Portugal, {cchristo@dem.ist.utl.pt, {duarte.valerio,mramalho,carlos.cardeira}@ist.utl.pt

F. Carreira and J. Calado are with IDMEC / Instituto Superior Técnico, Technical University of Lisbon and with Instituto Superior de Engenharia de Lisboa / IPL, R. Conselheiro Emídio Navarro 1, 1959-007 Lisboa, {fcarreira,jcalado}@dem.isel.ipl.pt

P. Oliveira is with ISR / Instituto Superior Técnico and IDMEC / Instituto Superior Técnico, Technical University of Lisbon, address as above, p.oliveira@dem.ist.utl.pt

that the robot may sense in the environment, which will subsequently be used for robot localization. In practice, given one environment, there is no guarantee that the same features will be present in the environment on subsequent visits of the robot to the same localization (loop closure problem). For instance, fast corners [37] are a very efficient way to detect features in an image but the number of corners actually found depends on many tuning parameters. Different corners may appear in different images taken from the same localization at different times. Random Sample Consensus (RANSAC) is considered the state of the art technique to keep track of features while disregarding outliers but in practice all these strategies rely on some structure of the environment [3], [23], [10].

This paper follows an alternative approach resorting to Principal Component Analysis (PCA) that actually does not depend on any predefined structure of the environment. There should always be something to distinguish data acquired in one location to data acquired in another location but no previous assumptions on the predefined structure of the environment needs to be considered. The PCA data analysis corresponds to the computation of the data orthogonal components that will make each dataset different. Hence, the localization is defined based on the PCA of the large amount of data taken from the unstructured environment.

### A. Current Practices

The use of vision systems for robot localization is very common [35], [34] due to the ability to obtain information about the environment. Many vision systems compute the robot pose (position and attitude) from features of the environment, either from the entire image [14], extracting lines [22], simply getting points of interest [16], [12], or extracting scale-invariant features [24]. The computational complexity of such algorithms to obtain features is not negligible, thus the implementation in real-time systems still demands the search for other approaches of reduced complexity.

Very successful implementations of visual odometry are presented in [34], where a robot was able to localize itself outdoors based on a minimum number of singular points that have to be present in the environment. Although many robots use cameras to look around itself to get its global pose in the environment [36], [12], [21], others use a single camera looking upward [16], [11], [39]. The use of vision from the ceiling has the advantage that images can be considered without scaling, i.e. a 2D image problem results and will be pursued in this work.

Ceiling based navigation resorts to a common problem of terrain based navigation. A commonly used synthesis technique for the terrain based navigation design problem has been Extended Kalman Filtering, see [15], [25], [33],

and the references therein. However, several authors reported instability and divergence problems of the proposed navigation systems, precluding their use in general. Correlation techniques have also been suggested to solve the problem at hand [5], [31], however with an high computational load required. To overcome these problems particle filters - discrete stochastic approximations to the optimal nonlinear filter - have also been developed [18], with a performance near the optimal, according with the Cramèr Rao lower bound obtained, but very computationally demanding. New mixed Kalman/particle filter methodologies have also been proposed [8], with similar performance to the classical particle filters, but where a large number of particles still must be used with the corresponding high computational burden.

### B. PCA-based localization and optimal estimation

Since feature based techniques are computationally heavy, some researchers have been working to find methods to make this process more efficient. To achieve reduced complexity algorithms, the use of PCA in mobile robots for self-localization has been explored [21], [26], [2]. However, all these approaches use front or omnidirectional cameras, causing the algorithms to address problems of occlusion or comparison with images in different planes. In [32], PCA was used for terrain reference navigation of underwater vehicles. The PCA-based localization system that we present is this work corresponds to a experimental validation of the one proposed in [32], using a Differential Drive Mobile Robot [7] equipped with a video camera looking upwards to the ceiling.

Beyond the problems of image processing for self-localization, another challenge is to deal with the fusion of the PCA-based position with the odometry data that is given by the robot kinematics. Mobile robot kinematics (e.g. Differential Drive) are, in general, non linear. This fact prevents the direct use of a Kalman Filter, which is a linear optimal estimator. To tackle this problem, many localization systems use the Extended Kalman Filter (EKF) with well characterized optimality and stability limitations. Even though it can give a reasonable performance, the EKF may diverge in consequence of wrong linearization or sensor noise.

In this paper, we use a Linear Parameter Varying (LPV) model for the Differential Drive Car, thus avoiding the non-linear model issues mentioned above. Moreover, the filter also estimates the slippage that is eventually present in the reality. Many researchers tend to neglect slippage: our approach addresses the problem explicitly. As slippage is inevitable, we append a state to our model to express the slippage explicitly. The filter estimates both angular slippage and robot localization. Furthermore, the optimal estimate is achieved, under the assumption that disturbance noise can be modeled by Gaussian distributions, with global stable error dynamics [32], [6].

### C. Advantages and drawbacks

The proposed PCA-based position sensor and localization estimation has the following advantages:

- The robot is able to self-locate in an indoor environment, only with on-board sensors (no external sensors or landmarks are required);

- The algorithm is fast, thus it consumes very few computational resources;
- The database of images stored on-board the mobile robot is of reduced size, when compared with the total number of images considered;
- The memory to allocate for the database storage is flexible and related with the required positioning error accuracy;
- No assumption is made about specific features in the environment: thus this system can operate in an unstructured environment where the only requirement is that images must be different in each location;
- Under Gaussian assumption for the disturbances, the localization system estimates in real time the position and angular slippage with global stable error dynamics.

Some of the limitation for the proposed approach include:

- The robots should work in buildings with ceilings where some information can be found (e.g. building-related systems such as HVAC, electrical and security systems, etc.);
- The ceilings should be static: the system cannot be used outdoors as the sky is far from static and changes randomly;
- The system is formulated in a digital discretized version as well as the PCA approach pursued.
- A general limitation of all vision-based systems is their sensitivity relative to lighting conditions. However, a mix of image and distance (e.g. using time of flight or structured light cameras) would increase the robustness to lighting conditions.

This paper is organized as follows: in section II, the principal component analysis technique is introduced in detail. In section III, mobile robot kinematics models are presented, including a linear model for the robot attitude and a LPV model for the robot position. Section IV shows a set of experimental results to validate and assess the performance of the proposed PCA-based positioning sensor and the linear Kalman filter based localization system. Future work is presented in section V, where a multi agent architecture is briefly outlined. Finally, in section VI some conclusions are drawn.

## II. PRINCIPAL COMPONENT ANALYSIS

In this section the fundamentals of the positioning system proposed in this work will be introduced. The proposed methodology resorts to optimal signal processing techniques, namely PCA, based on the Karhunen-Loève (KL) transform to obtain a nonlinear positioning sensor. Considering all linear transformations, PCA allows for the optimal approximation to a stochastic signal in the least squares sense. Furthermore, it is a well known signal expansion technique with uncorrelated coefficients for dimensionality reduction. These features make the KL transform interesting for many signal processing applications such as data compression, image and voice processing, data mining, exploratory data analysis, pattern recognition and time series prediction. For a thorough introduction to this topic and a number of state of the art applications see [17].

Consider a set of  $M$  stochastic signals  $\mathbf{x}_i \in \mathbb{R}^N$ ,  $i = 1, \dots, M$ , each corresponding to the stacked version of an

image acquired with the video camera installed onboard the mobile robot and represented as a column vector with mean  $\mathbf{m}_x = \frac{1}{M} \sum_{i=1}^M \mathbf{x}_i$ . The purpose of the KL transform is to find an orthogonal basis to decompose a stochastic signal  $\mathbf{x}$ , from the same original space, to be computed as  $\mathbf{x} = \mathbf{U}\mathbf{v} + \mathbf{m}_x$ , where vector  $\mathbf{v} \in \mathbb{R}^N$  is the projection of  $\mathbf{x}$  in the basis, i.e.  $\mathbf{v} = \mathbf{U}^T(\mathbf{x} - \mathbf{m}_x)$ . Matrix  $\mathbf{U} = [\mathbf{u}_1 \ \mathbf{u}_2 \ \dots \ \mathbf{u}_N]$  should be composed by the  $N$  orthogonal column vectors of the basis, verifying the eigenvalue problem

$$\mathbf{R}_{xx}\mathbf{u}_j = \lambda_j\mathbf{u}_j, \quad j = 1, \dots, N, \quad (1)$$

where  $\mathbf{R}_{xx}$  is the covariance matrix, computed from the set of  $M$  experiments using

$$\mathbf{R}_{xx} = \frac{1}{M-1} \sum_{i=1}^M (\mathbf{x}_i - \mathbf{m}_x)(\mathbf{x}_i - \mathbf{m}_x)^T. \quad (2)$$

Assuming that the eigenvalues are ordered, i.e.  $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_N$ , the choice of the first  $n \ll N$  principal components leads to an approximation to the stochastic signals given by the ratio on the covariances associated with the components, i.e.  $\sum_n \lambda_n / \sum_N \lambda_N$ . In many applications, where stochastic multidimensional signals are the key to overcome the problem at hand, this approximation can constitute a large dimensional reduction and thus a computational complexity reduction.

The advantages of PCA are threefold: i) it is an optimal (in terms of mean squared error) linear scheme for compressing a set of high dimensional vectors into a set of lower dimensional vectors; ii) the model parameters can be computed directly from the data (by diagonalizing the ensemble covariance); and iii) given the model parameters, projection into and from the bases are computationally inexpensive operations,  $\sim \mathcal{O}(nN)$ . These advantages suit specially our problem, as the computation power, energy and data storage on-board should be kept as reduced as possible to augment the operation interval and reduce the cost of the systems on-board.

Assume that scenario in the area of indoor mobile robotics (e.g. industrial automation or robotic office applications), where a navigation system to be installed on one or more mobile robots must be developed and operated. In this scenario it is considered that there is data available allowing to develop a positioning system that recognizes the actual position of the robot in real time. The steps to implement a PCA-based positioning sensor using this visual data will be outlined next.

Prior to the deployment of the robots, the visual data of the area under consideration should be partitioned in *mosaics* with fixed dimensions  $N_x$  by  $N_y$ . After reorganizing this two-dimensional data in vector form, e.g. stacking the columns, a set of  $M$  stochastic signals  $\mathbf{x}_i \in \mathbb{R}^N$ ,  $N = N_x N_y$  results. The number of signals  $M$  to be considered depends on the mission scenario and on mosaic overlapping. The KL transform can be computed, using equations (1) and (2); the eigenvalues must be ordered and the number  $n$  of the principal components to be used should be selected, according with the required level of approximation.

The following data should be recorded for later use:

- 1) the data ensemble mean  $\mathbf{m}_x$ ;

- 2) the matrix transformation with  $n$  eigenvectors

$$\mathbf{U}_n = [\mathbf{u}_1 \ \dots \ \mathbf{u}_n]; \quad (3)$$

- 3) the projection on the selected basis of all the mosaics, computed using

$$\mathbf{v}_i = \mathbf{U}_n^T(\mathbf{x}_i - \mathbf{m}_x), \quad i = 1, \dots, M; \quad (4)$$

- 4) the coordinates of the center of the mosaics

$$(x_i, y_i), \quad i = 1, \dots, M. \quad (5)$$

During the mission, at the time instants  $t_k = Lk$  (where  $L$  is a positive integer), the acquired images will constitute the input signal  $\mathbf{x}$  to the PCA positioning system. The following tasks should be performed:

- a) compute the projection of the signal  $\mathbf{x}$  into the basis, using

$$\mathbf{v} = \mathbf{U}_n^T(\mathbf{x} - \mathbf{m}_x); \quad (6)$$

- b) given an estimate of the current horizontal coordinates of the robot position  $\hat{x}$  and  $\hat{y}$ , provided by the navigation system, search on a given neighborhood  $\delta$  the mosaic that verifies

$$\forall_i \|\hat{x} \hat{y}^T - [x_i \ y_i]^T\|_2 < \delta, \quad r_{\text{PCA}} = \min_i \|\mathbf{v} - \mathbf{v}_i\|_2; \quad (7)$$

- c) given the mosaic  $i$  which is closest to the present input, its center coordinates  $(x_i, y_i)$  will be selected as the  $x_m$  and  $y_m$  measurements.

The relation  $\mathbf{f}$  between  $r_{\text{PCA}}$  and the positioning sensor error covariance  $\mathbf{R}$  (observation noise) to be used in the  $\mathcal{H}_2$  estimation problem

$$\mathbf{R} = \mathbf{f} r_{\text{PCA}} \quad (8)$$

will be chosen according to the chosen environment. Note that the image-based PCA positioning system described above can be straightforwardly extended to incorporate data from other sensors installed onboard mobile robots such as magnetometers and range information from time-of-flight cameras or structured-light 3D scanners (e.g. Microsoft Kinect).

### III. MODEL

This section will present the differential mobile robot platform and the proposed linear model Differential Drive Car implemented in the developed positioning system.

The experimental validation of the proposed positioning system was performed resorting to a low cost mobile robotic platform [7], with the configuration of a Differential Drive Car. This platform has a PC laptop that controls the motors through a closed loop motor controller connected by a USB emulated serial port (see figures 1 and 2). The low replication cost for these platforms will be instrumental during the future tasks envisioned relying on cooperation and multi-agent systems (see section V).

For the purpose of this work the platform was adapted to include:

- a camera, pointing upwards to the ceiling;
- a compass, located in an extension arm to avoid the motors' magnetic interference (see figure 2).



Fig. 1. Mobile robot validation platforms

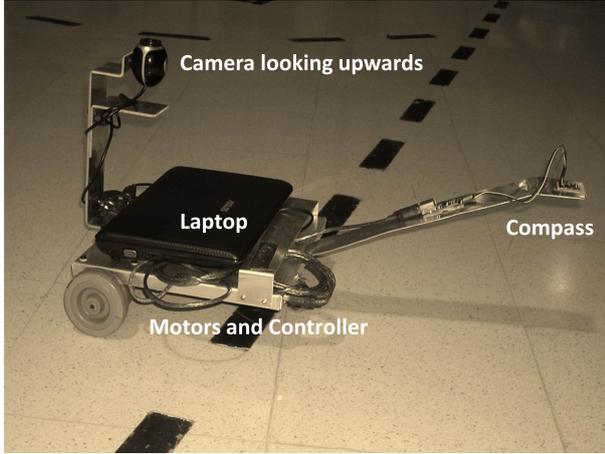


Fig. 2. Mobile platform equipped with camera and compass

The classical differential drive mobile robot model is given by

$$\dot{x} = u \cos \theta \quad (9)$$

$$\dot{y} = u \sin \theta \quad (10)$$

$$\dot{\theta} = \omega \quad (11)$$

where

- $u$  is the common mode speed,
- $x$  and  $y$  are the robot coordinates in the world referential,
- $\theta$  is the orientation angle of the robot in the world referential, and
- $\omega$  is the angular speed.

#### A. Differential Drive Car Discrete Model

The continuous model defined by equations (9)–(11) is nonlinear, which is a strong limitation for the use of a Kalman filter. However, the model can be rewritten, increasing the state space and choosing carefully new state variables, so that the nonlinear system becomes a Linear Parameter Varying (LPV) model for the Differential Drive Car, as shown in this subsection. The implicit assumption is that the localizer is processed in a discrete processor where  $u$  and  $\omega$  remain constant (zero order hold assumption) between two consecutive processing times. This is a

reasonable approximation in a discrete controller, assuming that the robot dynamics are fully absorbed by the controller.

Thus, differentiating the equations (9)–(11):

$$\ddot{x} = -u \omega \sin \theta = -\omega \dot{y} \quad (12)$$

$$\ddot{y} = u \omega \cos \theta = \omega \dot{x} \quad (13)$$

$$\ddot{\theta} = \dot{\omega} \quad (14)$$

and choosing as state vector for Differential Drive Car:

$$\mathbf{x} = [x \quad \dot{x} \quad y \quad \dot{y}]^T \quad (15)$$

the nonlinear model could be rewritten as the LPV model:

$$\dot{\mathbf{x}} = \underbrace{\begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -\omega \\ 0 & 0 & 0 & 1 \\ 0 & \omega & 0 & 0 \end{bmatrix}}_{\mathbf{A}(\omega)} \mathbf{x} + \boldsymbol{\mu} \quad (16)$$

$$\dot{\theta} = \omega \quad (17)$$

$$\mathbf{y} = \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}}_{\mathbf{C}} \mathbf{x} + \boldsymbol{\gamma} \quad (18)$$

#### B. Discretization

Considering the LPV model of the Differential Drive Car is processed in a digital processor, and assuming  $\omega$  to be constant between a sampling time (zero order hold assumption), the solution (16)–(17) with state vector (15) is given by:

$$\mathbf{x}(t) = e^{\mathbf{A}(\omega)t} \mathbf{x}(0) + \int_0^t e^{\mathbf{A}(\omega)(t-\tau)} \mathbf{B} d\tau \cdot \boldsymbol{\mu}(t) \quad (19)$$

solving the equation (19) by parts:

$$\begin{aligned} e^{\mathbf{A}(\omega)t} &= \exp \left( \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -\omega \\ 0 & 0 & 0 & 1 \\ 0 & \omega & 0 & 0 \end{bmatrix} t \right) = \\ &= \begin{bmatrix} 1 & \frac{i e^{-i\omega t}}{2\omega} - \frac{i e^{i\omega t}}{2\omega} & 0 & -\frac{1}{\omega} + \frac{e^{-i\omega t}}{2\omega} + \frac{e^{i\omega t}}{2\omega} \\ 0 & \frac{e^{-i\omega t}}{2} - \frac{e^{i\omega t}}{2} & 0 & -\frac{i e^{-i\omega t}}{2} + \frac{i e^{i\omega t}}{2} \\ 0 & \frac{1}{\omega} - \frac{e^{-i\omega t}}{2\omega} - \frac{e^{i\omega t}}{2\omega} & 1 & \frac{i e^{-i\omega t}}{2\omega} - \frac{i e^{i\omega t}}{2\omega} \\ 0 & \frac{i e^{-i\omega t}}{2} - \frac{i e^{i\omega t}}{2} & 0 & \frac{e^{-i\omega t}}{2} - \frac{e^{i\omega t}}{2} \end{bmatrix} = \\ &= \begin{bmatrix} 1 & \frac{\sin \omega t}{\omega} & 0 & \frac{1}{\omega} + \frac{\cos \omega t}{\omega} \\ 0 & \cos \omega t & 0 & -\sin \omega t \\ 0 & \frac{1}{\omega} - \frac{\cos \omega t}{\omega} & 1 & \frac{\sin \omega t}{\omega} \\ 0 & \sin \omega t & 0 & \cos \omega t \end{bmatrix} \quad (20) \end{aligned}$$

and taking into account that  $\mathbf{B} = \mathbf{I}$ :

$$\begin{aligned}
& \int_0^t e^{\mathbf{A}(\omega)(t-\tau)} \mathbf{B} d\tau = \\
& = \int_0^t \exp \left( \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -\omega \\ 0 & 0 & 0 & 1 \\ 0 & \omega & 0 & 0 \end{bmatrix} (t-\tau) \right) \mathbf{I} d\tau = \\
& = \int_0^t \begin{bmatrix} 1 & \frac{\sin(\omega(t-\tau))}{\omega} & 0 & -\frac{1}{\omega} + \frac{\cos(\omega(t-\tau))}{\omega} \\ 0 & \cos(\omega(t-\tau)) & 0 & -\sin(\omega(t-\tau)) \\ 0 & \frac{1}{\omega} - \frac{\cos(\omega(t-\tau))}{\omega} & 1 & \frac{\sin(\omega(t-\tau))}{\omega} \\ 0 & \sin(\omega(t-\tau)) & 0 & \cos(\omega(t-\tau)) \end{bmatrix} \cdot (t-\tau) d\tau = \\
& = \begin{bmatrix} \tau & \frac{\cos(\omega(t-\tau))}{\omega^2} & 0 & -\frac{\omega\tau + \sin(\omega(t-\tau))}{\omega^2} \\ 0 & -\frac{\sin(\omega(t-\tau))}{\omega} & 0 & -\frac{\cos(\omega(t-\tau))}{\omega} \\ 0 & \frac{\omega\tau + \sin(\omega(t-\tau))}{\omega^2} & \tau & \frac{\cos(\omega(t-\tau))}{\omega} \\ 0 & \frac{\cos(\omega(t-\tau))}{\omega} & 0 & -\frac{\sin(\omega(t-\tau))}{\omega} \end{bmatrix} \Bigg|_0^t = \\
& = \begin{bmatrix} t & \frac{1-\cos \omega t}{\omega^2} & 0 & -\frac{\omega t - \sin \omega t}{\omega^2} \\ 0 & \frac{\sin \omega t}{\omega} & 0 & -\frac{1-\cos \omega t}{\omega} \\ 0 & \frac{\omega t - \sin \omega t}{\omega^2} & t & \frac{1-\cos \omega t}{\omega^2} \\ 0 & \frac{1-\cos \omega t}{\omega} & 0 & \frac{\sin \omega t}{\omega} \end{bmatrix} \quad (21)
\end{aligned}$$

Replacing (20)–(21) in (19), the discrete model of Differential Drive Car can be defined by the LPV in terms of angular speed of robot  $\omega$ :

$$\begin{aligned}
\mathbf{x}(k+1) &= \underbrace{\begin{bmatrix} 1 & \frac{\sin \omega T}{\omega} & 0 & \frac{1}{\omega} + \frac{\cos \omega T}{\omega} \\ 0 & \cos \omega T & 0 & -\sin \omega T \\ 0 & \frac{1}{\omega} - \frac{\cos \omega T}{\omega} & 1 & \frac{\sin \omega T}{\omega} \\ 0 & \sin \omega T & 0 & \cos \omega T \end{bmatrix}}_{\mathbf{A}(\omega(k))} \mathbf{x}(k) + \\
&+ \underbrace{\begin{bmatrix} T & \frac{1-\cos \omega T}{\omega^2} & 0 & -\frac{\omega T - \sin \omega T}{\omega^2} \\ 0 & \frac{\sin \omega T}{\omega} & 0 & -\frac{1-\cos \omega T}{\omega} \\ 0 & \frac{\omega T - \sin \omega T}{\omega^2} & T & \frac{1-\cos \omega T}{\omega^2} \\ 0 & \frac{1-\cos \omega T}{\omega} & 0 & \frac{\sin \omega T}{\omega} \end{bmatrix}}_{\mathbf{G}(\omega(k))} \boldsymbol{\mu}(k) \quad (22)
\end{aligned}$$

$$\mathbf{y}(k) = \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}}_{\mathbf{C}} \mathbf{x}(k) + \boldsymbol{\gamma}(k) \quad (23)$$

Once that for  $\omega(k) = 0$  some expressions results in an indetermination, in this case it is applied:

$$\mathbf{A}(\omega)(k) = \lim_{\omega \rightarrow 0} \mathbf{A}(\omega(k)) = \begin{bmatrix} 1 & T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (24)$$

$$\mathbf{G}(\omega)(k) = \lim_{\omega \rightarrow 0} \mathbf{G}(\omega(k)) = \begin{bmatrix} T & \frac{T^2}{2} & 0 & 0 \\ 0 & T & 0 & 0 \\ 0 & 0 & T & \frac{T^2}{2} \\ 0 & 0 & 0 & T \end{bmatrix} \quad (25)$$

### C. Position Optimal Estimation

The position of the Differential Drive Car can be estimated by a Kalman Filter based on the PCA sensor position measurements  $[x_{\text{PCA}}(k) \ y_{\text{PCA}}(k)]^T$  (see section II) and

the LPV model ( $\mathbf{A}(\omega(k))$ ) presented in above subsection. To implement this position estimator we assume that the error in comom-mode actuation and in the PCA position sensor is a zero-mean uncorrelated Gaussian noise  $\boldsymbol{\mu} \sim N(0, \sigma_i^2)$  and  $\boldsymbol{\gamma} \sim N(0, \sigma_i^2)$ , respectively. Applying the usual loop with a Kalman gain to fuse the PCA position sensor with the LPV model, the position estimator will be implemented using the equation (26), that is represented by the block diagram of figure 3:

$$\begin{aligned}
\hat{\mathbf{x}}(k+1) &= \mathbf{A}(\omega(k))\hat{\mathbf{x}}(k) + \\
&\mathbf{K}(k)([x_{\text{PCA}}(k) \ y_{\text{PCA}}(k)]^T - \mathbf{C}\hat{\mathbf{x}}(k)) \quad (26)
\end{aligned}$$

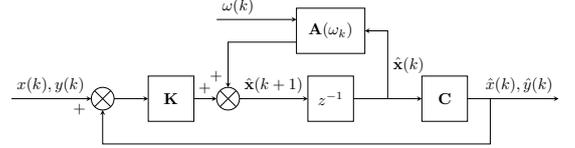


Fig. 3. Block diagram of position estimator

Considering the implementation of an LPV model with uncorrelated white noise, the position estimator is, under the assumption of the Kalman filter properties, an optimal estimator. So, the position estimated performing a zero-mean error and global stabilization considering any wrong initial conditions. This is an important consideration because usually, to estimate the position of Differential Drive Car, is common to apply EKF, which position error is not zero-mean and can unstabilize.

Applying the usual approach of Kalman filter design, the covariance of the prediction estimation error for the LPV model presented in (22) and considering the process covariance in comon-mode actuation  $\mathbf{Q}_1$  is given by:

$$\mathbf{P}(k) = \mathbf{A}(\omega(k))\mathbf{P}(k)\mathbf{A}(\omega(k))^T + \mathbf{G}(\omega(k))\mathbf{Q}_1\mathbf{G}(\omega(k))^T \quad (27)$$

To obtain the covariance of the prediction estimation error, the Kalman filter starts its estimation considering that the environment is uniform, i.e. the initial error covariance  $\mathbf{P}(0)$  is equal in  $x$  and  $y$  directions. The error covariance in the linear actuation  $\mathbf{Q}_1$ , was obtained performing tests based on linear trajectories, and is also equal in both directions. Following the classical Kalman filter approach, the error covariance was updated with the uncertainty of the the position measured:

$$\mathbf{P}(k) = \mathbf{P}(k) - \mathbf{P}(k)\mathbf{C}^T(\mathbf{C}\mathbf{P}(k)\mathbf{C}^T + \mathbf{R}(k))^{-1}\mathbf{C}\mathbf{P}(k) \quad (28)$$

where  $\mathbf{R}(k)$  is the the observation uncertainty for the instant  $k$  based in the uncertainty of the PCA position sensor and  $\mathbf{R}_{\text{xy}}$  the observation noise described in equation (8):

$$\mathbf{R}(k) = \mathbf{R}_{\text{xy}} \cdot \mathbf{f} \ r_{\text{PCA}}(k) \quad (29)$$

The error covariance of the PCA positioning sensor ( $\mathbf{R}_{\text{xy}}$ ) has been calculated with the position error in a predefined trajectory and is not equal in  $x$  and  $y$  directions because, in reality, the environment is not identical in both direction

resulting in different uncertainty measurement of the sensor in each direction.

Considering the uncertainty of the estimator  $\mathbf{P}(k)$  and in the PCA position sensor  $\mathbf{P}(k)$  the Kalman gain that is apply in the global position estimator for instant  $k$  is obtained:

$$\mathbf{K}(k) = \mathbf{P}(k)\mathbf{C}^T\mathbf{R}(k)^{-1} \quad (30)$$

The Kalman gain  $\mathbf{K}$  is a dynamic gain proportional to the estimation uncertainty and inversely proportional of measurement uncertainty in the PCA position sensor. Considering that this gain is applied to a LPV model with a white noise in the actuation and in the position sensor, the results of the position estimated is always convergent and stable for any initial conditions in the state vector.

#### D. Attitude Optimal Estimation

In addition with the position estimation, the self-localization system is completed with the estimation of the attitude of the robot. As metioned above, due the uncertainties of measures (radius of wheels, unknowdge of contact points), the angular slippage is inevitable in a Differential Drive Car. Thus, the proposed attitude estimator will projected to estimate both, the attitude of the robot (and consequently its angular speed) and the angular slippage caused by the mentioned uncertainties.

The kinematic model that describes the attitude, tacking into account the angular slippage is:

$$\dot{\psi} = \omega + s + \mu_1 \quad (31)$$

$$\dot{s} = 0 + \mu_2 \quad (32)$$

where the the following assumptions were considered:

- the angular slippage is constant or slowly varying (i.e.  $\dot{s} = 0$ );
- the noise in the actuation (motors are in closed loop) and the slippage velocity are assumed as zero-mean uncorrelated white Gaussian noise,  $\mu_i \sim N(0, \sigma_i^2)$ .

Expressing the equation (32) in a state-space with  $\boldsymbol{\theta} = [\psi \ s]^T$ , the model dynamics is defined by

$$\dot{\boldsymbol{\theta}} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \boldsymbol{\theta} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} \omega + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix} \quad (33)$$

$$y(k) = [1 \ 0] \boldsymbol{\theta}(k) + \gamma(k) \quad (34)$$

Since the attitude estimator is processed in a digital processor, the discrete model is obtained assuming that the angular speed  $\omega$  is constant (zero order hold assumption) between two consecutive processing times, the solution of this model is given by:

$$\boldsymbol{\theta}(t) = e^{\mathbf{A}t} \boldsymbol{\theta}(0) + \int_0^t e^{\mathbf{A}(t-\tau)} \mathbf{B} d\tau \omega \quad (35)$$

Taking into account that

$$e^{\mathbf{A}t} = \exp\left(\begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} t\right) = \begin{bmatrix} 1 & t \\ 0 & 1 \end{bmatrix} \quad (36)$$

and

$$\begin{aligned} & \int_0^t e^{\mathbf{A}(t-\tau)} \mathbf{B} d\tau \omega = \\ & = \int_0^t \exp\left(\begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} (t-\tau)\right) \begin{bmatrix} 1 \\ 0 \end{bmatrix} d\tau \omega = \\ & = \int_0^t \begin{bmatrix} 1 & t-\tau \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} d\tau \omega = \\ & = \begin{bmatrix} \tau \\ 0 \end{bmatrix} \Big|_0^T \omega = \begin{bmatrix} T \\ 0 \end{bmatrix} \omega \end{aligned} \quad (37)$$

the discrete model of attitude for a sampling time  $T$  is obtained replacing (36)–(37) in (35):

$$\boldsymbol{\theta}(k+1) = \underbrace{\begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}}_{\mathbf{A}} \boldsymbol{\theta}(k) + \underbrace{\begin{bmatrix} T \\ 0 \end{bmatrix}}_{\mathbf{B}} \omega(k) \quad (38)$$

$$y(k) = \underbrace{\begin{bmatrix} 1 & 0 \end{bmatrix}}_{\mathbf{C}} \boldsymbol{\theta}(k) \quad (39)$$

Applying a Kalman filter for this discrete model, the estimator vector is given by:

$$\hat{\boldsymbol{\theta}}(k+1) = \mathbf{A}\hat{\boldsymbol{\theta}}(k) + \mathbf{B}\omega(k) + \mathbf{K}[y(k) - \mathbf{C}\hat{\boldsymbol{\theta}}(k)] \quad (40)$$

Uncoupling the matrix equation (40), and considering that the output state ( $y(k)$ ) is  $\psi(k)$ , we obtain:

$$\hat{\psi}(k+1) = \hat{\psi}(k) + T\hat{s}(k) + T\omega_0(k) + K_1(\psi(k) - \hat{\psi}(k)) \quad (41)$$

$$\hat{s}(k+1) = \hat{s}(k) + K_2(\psi(k) - \hat{\psi}(k)) \quad (42)$$

where:

- $\hat{\psi}$  is the estimated attitude of the robot
- $\hat{s}$  is the estimated angular slippage (caused by the uncertainty in the robot model, e.g. non-nominal dimensions of wheels).
- $\omega_0(k)$  is the angular speed of the robot:

The angular speed of the Differential Drive Car of a sampling time  $T$  is given by:

$$\omega_0(k) = \frac{(\alpha_r(k) - \alpha_r(k-1)) - (\alpha_l(k) - \alpha_l(k-1))}{T \cdot l} \quad (43)$$

where:

- $l$  is the distance between wheels
- $\alpha_r$  is the angle of the right wheel
- $\alpha_l$  the angle of the left wheel.
- $\omega_o(k)$  is the angular speed given by the odometry of the robot

This model, based in the numerical difference of odometric readings, is represented by the block diagram of figure 4.

To avoid obtaining the attitude of robot through a numerical difference (that amplifies the noise of the angular sensor), and since it is possible to obtain the angles of the wheels ( $\alpha_r$  and  $\alpha_l$ ) directly from the hardware the differential attitude can be calculated by:

$$\psi_o(k) = (\alpha_r - \alpha_l) \frac{r}{l} \quad (44)$$

where

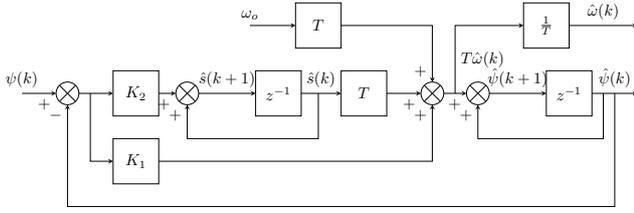


Fig. 4. Block diagram of attitude estimator with numerical difference

- $r$  is the radius of the wheels
- $l$  the distance between wheels.
- $\psi_o(k)$  is the attitude given by the odometry of the robot

The block diagram of Figure 4 should be redesigned to the equivalent system in Figure 5, and the model of attitude estimator (41)–(42) can be rewritten as

$$\hat{\psi}(k) = \beta(k) + \psi_o(k) \quad (45)$$

$$\beta(k+1) = \beta(k) + T\hat{s}(k) + K_1(\psi(k) - \hat{\psi}(k)) \quad (46)$$

$$\hat{s}(k+1) = \hat{s}(k) + K_2(\psi(k) - \hat{\psi}(k)) \quad (47)$$

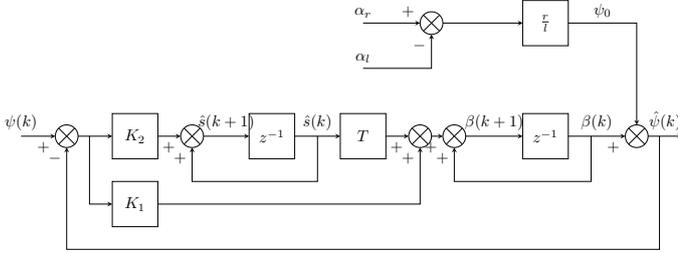


Fig. 5. Block diagram of attitude estimator without numerical difference

The angular speed of the robot that will be used in LPV (23) is obtained through a numerical difference of the estimated attitude of the robot:

$$\hat{\omega}(k) = \frac{\hat{\psi}(k) - \hat{\psi}(k-1)}{T} \quad (48)$$

This solution, although calculated with a numerical difference, is filtered by the attitude estimator, causing less noise than the direct numerical difference from the encoders.

#### E. Observability of the Global Self-localization Models

To verify that the proposed models could be implemented in a localization system and to know the possible information that can be obtained through it, an observability study to models must be also realized.

Thus, to verify if the model of attitude and the angular slippage is observable, the matrix of (38)–(39) was considered to obtain the observability matrix:

$$\mathcal{O}_a = \begin{bmatrix} \mathbf{C} \\ \mathbf{CA} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (49)$$

which results in a  $\text{rank}(\mathcal{O}_a) = 2$ . Considering that  $\text{rank}(\mathcal{O}_a)$  is equal to the dimension of the state vector, it is possible to conclude that the two states are observable and is possible to implement this model estimate attitude of the robot and the angular slippage.

Repeating the same process to study the observability of the LPV model presented in equations (22)–(23), the observability matrix is given by:

$$\mathcal{O}_p = \begin{bmatrix} \mathbf{C} \\ \mathbf{CA}(\omega(k)) \\ \mathbf{CA}(\omega(k))^2 \\ \mathbf{CA}(\omega(k))^3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -\omega \\ 0 & \omega & 0 & 0 \\ 0 & -\omega^2 & 0 & 0 \\ 0 & 0 & 0 & -\omega^2 \end{bmatrix} \quad (50)$$

which verifies  $\text{rank}(\mathcal{O}_p) = 4$ , allowing to concluded that the model is observable and that it is possible be implemented to estimate the position of the robot.

Finally, in addition to position observability study presented above, and in order to verify the possibility of common mode slippage estimation, the model of differential drive (9)–(11) will be rewritten as

$$\dot{x} = (u + e) \cos \theta \quad (51)$$

$$\dot{x} = (u + e) \sin \theta \quad (52)$$

$$\dot{\theta} = \omega \quad (53)$$

$$\dot{e} = 0 \quad (54)$$

where  $e$  is the common mode slippage, constant between two sampling times (zero order hold assumption). Differentiating (51)–(54)

$$\begin{aligned} \ddot{x} &= \dot{u} \cos \theta - u\omega \sin \theta + \dot{e} \cos \theta - e\omega \sin \theta \\ &= -(u + e) \sin \theta = -\omega \dot{y} \end{aligned} \quad (55)$$

$$\begin{aligned} \dot{x} &= \dot{u} \sin \theta - u\omega \cos \theta + \dot{e} \sin \theta - e\omega \cos \theta \\ &= (u + e) \cos \theta = \omega \dot{x} \end{aligned} \quad (56)$$

$$\ddot{\theta} = \dot{\omega} \quad (57)$$

Choosing as state vector

$$\mathbf{x} = [x \quad \dot{x} \quad y \quad \dot{y} \quad e]^T \quad (58)$$

the model becomes:

$$\dot{\mathbf{x}} = \underbrace{\begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -\omega & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & \omega & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}}_{\mathbf{A}(\omega(k))} \mathbf{x} \quad (59)$$

$$\mathbf{y} = \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}}_{\mathbf{C}} \mathbf{x} \quad (60)$$

From the dynamic and observable matrixes the observability

matrix may be obtained:

$$\mathcal{O}_p = \begin{bmatrix} \mathbf{C} \\ \mathbf{CA}(\omega(k)) \\ \mathbf{CA}(\omega(k))^2 \\ \mathbf{CA}(\omega(k))^3 \\ \mathbf{CA}(\omega(k))^4 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -\omega & 0 \\ 0 & \omega & 0 & 0 & 0 \\ 0 & -\omega^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & -\omega^2 & 0 \\ 0 & 0 & 0 & \omega^3 & 0 \\ 0 & -\omega^3 & 0 & 0 & 0 \end{bmatrix} \quad (61)$$

which verifies  $\text{rank}(\mathcal{O}_p) = 4$ . Once that the value of the  $\text{rank}(\mathcal{O}_p)$  is less than the dimension of the state vector, it is only possible to conclude that the system is not observable; in other words, the propose model allow the estimation of the angular slippage but does not the estimation of the common mode slippage of the robot.

#### IV. EXPERIMENTAL RESULTS

##### A. Environment

The mobile robot self-localization methodology proposed in this paper has been tested for the aforementioned mobile robot. To create the PCA eigenspace (the image database referred in the previous sections), the environment has been mapped with images captured in the same direction and referenced in a grid (see figure 6).

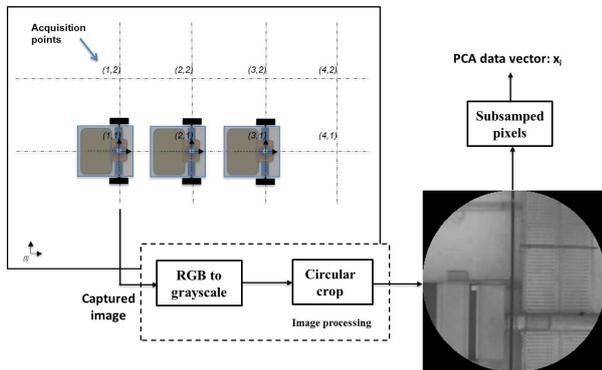


Fig. 6. Grid map and image processing to create a PCA eigenspace.

The covered area was a  $5\text{ m} \times 4.5\text{ m}$  surface rectangle, with a distance of  $0.3\text{ m}$  (in  $x$  and  $y$  axis) between captured images. The gray scale images were cropped with a circular mark (see image processing in figure 1) to allow the rotation of captured images. The pixels representing the images inside the circle have been converted into a vector. In order to compress the amount of processing data with negligible information loss, the mentioned vectors have been sampled considering a compression ratio of  $1/10$  and, thus, creating the PCA eigenspace (figure 6). Therefore, analyzing the corresponding PCA eigenvalues and selecting components that explain the images variability in an excess of  $85\%$ , results on an eigenspace (image database) of 22 eigenvectors. Considering the covariance's  $\mathbf{Q} = \mathbf{Q}(k)$  and  $\mathbf{R} = \mathbf{R}(k)$  used in the Kalman filter design, which are respectively the error covariance in the actuation system and in the sensors, some

measurement tests were performed with a sampled frequency of  $5\text{ Hz}$ . The error covariance in the differential actuation is represented by  $\mathbf{Q}_\omega$ , while the variable  $\mathbf{Q}_1$  stands for the error covariance in the common-mode actuation. For the measurement uncertainty,  $\mathbf{R}_\psi$  stands for the error covariance in the attitude sensor, while  $\mathbf{R}_{xy}$  stands for the error covariance in the position sensor. Thus, to parameterize the attitude estimator, the value  $\mathbf{Q}_\omega = 2.44 \times 10^{-5} (\text{rad} \cdot \text{s}^{-1})^2$  was obtained measuring the error covariance in the differential actuation along predefined circulars trajectories with different radius and  $\mathbf{R}_\psi = 9.4 \times 10^{-3} \text{ rad}^2$  being the covariance error of digital compass, has been measured using the same type of trajectories. For the position estimator, the error covariance in the linear actuation  $\mathbf{Q}_1 = 4.1 \times 10^{-6} \text{ m}^2$ , was obtained performing tests based on linear trajectories, while the error covariance of the PCA positioning sensor, has been calculated with the position error in a predefined trajectory being  $\mathbf{R}_{xy} = \text{diag}([1.6310^{-2} \quad 8.34 \times 10^{-1}]) \text{ m}^2$ .

##### B. Architecture

The proposed sensor consists of two Kalman filters and one PCA positioning sensor (figure 7), where, only with on-board sensors (camera, compass and encoders), is possible to estimate the robot attitude and position, as well as the angular motion speed and the robot angular slippage.

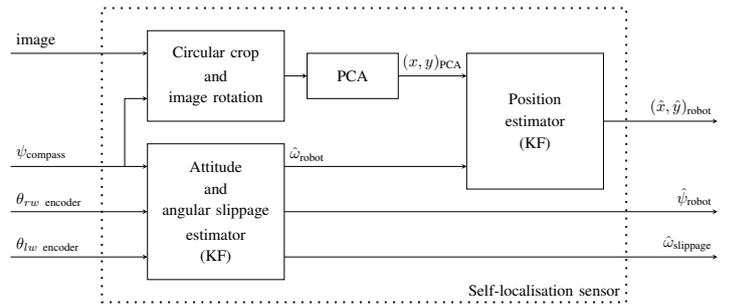


Fig. 7. Architecture of the self-localization sensor.

The following notation is used in figure 7:

- $\psi_{compass}$  is the orientation angle given by the compass
- $\theta_{rw\ encoder}$  is the angle given by the encoder of the right wheel
- $\theta_{lw\ encoder}$  is the angle given by the encoder of the left wheel
- $(x, y)_{PCA}$  is the coordinates given by the PCA sensor
- $(\hat{x}, \hat{y})_{robot}$  is the estimated robot coordinates in the world referential
- $\hat{\omega}_{robot}$  is the estimated angular speed
- $\hat{\omega}_{slippage}$  is the estimated differential slippage.

Detailing the architecture of the self-localization sensor, the Kalman filter depicted on the left of figure 7 implements the attitude optimal estimator model described above (Section III) and is responsible to estimate the mobile robot attitude and the differential slippage.

In order to apply the PCA algorithm described in Section II, the captured images should be rotated for zero degrees of attitude. The reason for such an operation is to achieve the images in the same orientation of the images stored in the PCA eigenspace. The image rotation has been performed

using the data acquired from the digital compass. However, before the image rotation has been performed, the image should have the same pre-processing treatment given to the images stored in the PCA eigenspace. In other words, the captured images should be converted from RGB (Red-Green-Blue color mode) to greyscale and cropped with a circular shape (see figure 6). The PCA block determines which image in the eigenspace has the better fits (eigenvector nearer to the captured image) with the acquired image and returns the coordinates of its centre (see Section II). Applying the proposed LPV model (see Section III) as a function of the estimated angular speed together with the coordinates given by the PCA position sensor, the self-localization sensor estimates the robot global position.

### C. Estimation on a Lawn-Mower trajectory

To test the mobile robot self-localization optimal estimator the proposed system was tested with the classical Lawn-Mower type trajectories. These trajectories have the advantage of combining both straight lines with curves allowing testing the localization system robustness under different conditions. Figure 8 depicts the results for a Lawn-Mower trajectory under the following conditions:

- Start robot position:  $x = 3.6$  m;  $y = 2.4$  m;  $\psi = 90^\circ$
- Robot speed:  $0.1$  m  $\cdot$  s $^{-1}$
- Sample frequency: 5 Hz
- Lawn-Mower trajectory with 4 turns
- Initial Kalman Filter estimation is the same as the initial robot position

To analyse the real error of the optimal estimator while robot was moving along the trajectory, same points were marked on the ground along the path followed by the robot.

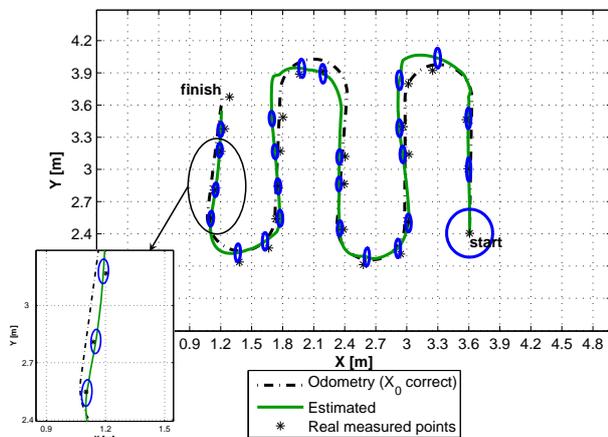


Fig. 8. Results for a Lawn-Mower trajectory (map 2D).

Analysing the results of the ground truth test (figure 8) it is possible to conclude that the estimator gives a very good approximation of the real trajectory. Making a trajectory zoom, in the end of the final straight line trajectory, it can be observed that the estimator uncertainty corresponding to the marks performed in the floor is lower than the position obtained through the robot's odometry. Sometimes, when the robot turns, the error associated with the value given by the compass origins a perturbation in the image rotation before

it being used in the PCA algorithm. Hence, due to such an effect the PCA algorithm estimates a position with one grid position of error ( $0.3$  m) as can be observed in figure 9. However, although these variations in the PCA position sensor could be observed, the estimator is able to recover (see figures 8 and 9).

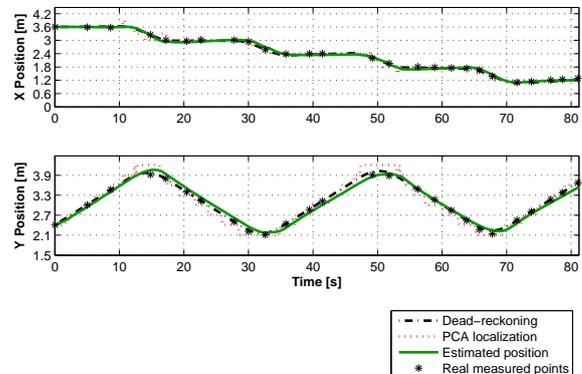


Fig. 9. Results for a Lawn-Mower trajectory (estimated position in time).

It is also worth measuring the statistical distribution of the errors in the position estimation, because according to the assumptions mentioned above, the distribution of the errors should be approximated by a zero mean Gaussian distribution. Figure 10 shows the histogram of the position errors for the coordinates  $x$  and  $y$ . The approximation to a zero mean Gaussian distribution is not exact, nevertheless the non-zero mean is explained by the fact that the trajectory is not random and due to some perturbations given by the PCA position sensor (that has a resolution of  $0.3$  m). Those perturbations are more visible in  $y$  estimated position where the PCA sensor has more perturbations when the robot curves (see figure 9). However, considering that the PCA position sensor resolution is  $0.3$  m, the non-zero mean,  $\bar{x} = -2.5 \times 10^{-3}$  m and  $\bar{y} = -1.64 \times 10^{-2}$  m, are very close to zero.

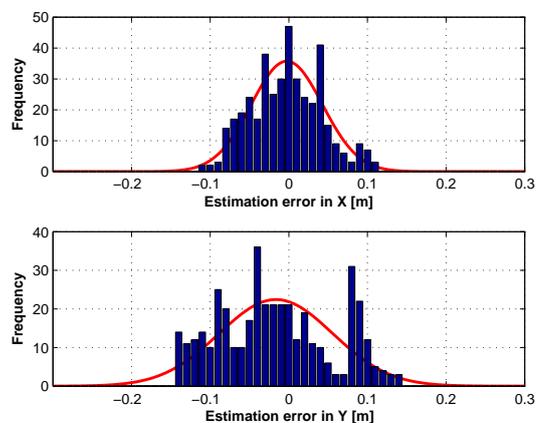


Fig. 10. Histogram of the estimated position error.

Figures 11 to 14 depict, respectively, the stabilization of the Kalman gains and the estimators' covariance in function of time. It can be observed that all estimators are convergent and fast to stabilize the uncertainties. This quickness to stabilize the position or the attitude is more visible in  $x$  position because, due to the characteristics of the PCA position sensor, the corresponding covariance is smallest than the covariance in  $y$  position.

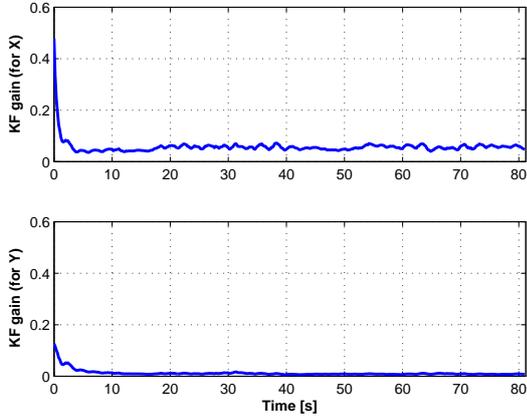


Fig. 11. Kalman gain evolution for position estimators.

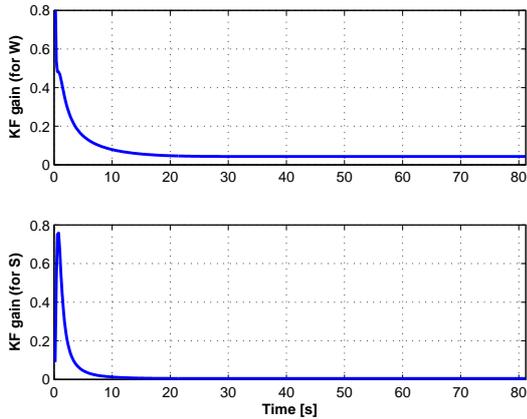


Fig. 12. Kalman gain evolution for attitude estimators.

As can be observed in the presented results the localization system is always stable performing a good estimation of the localization in spite of the low precision sensor used to measure the position looking up to the ceiling. An odometry base localization system would necessary diverge over time. A suitable data sensor fusion in the Kalman filters allows achieving an accurate sensor with a performance much higher than the performance of each individual sensor. In the previous tests can be observed that during the curves the system loses some precision, which is explained by the inevitable errors presented in the information provided by the compass and the consequent disturbance in the image rotation that will be compared with the images in the PCA eigenspace.

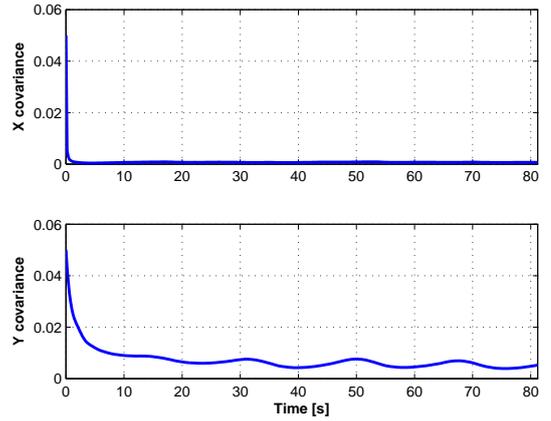


Fig. 13. Covariance evolution for position estimators.

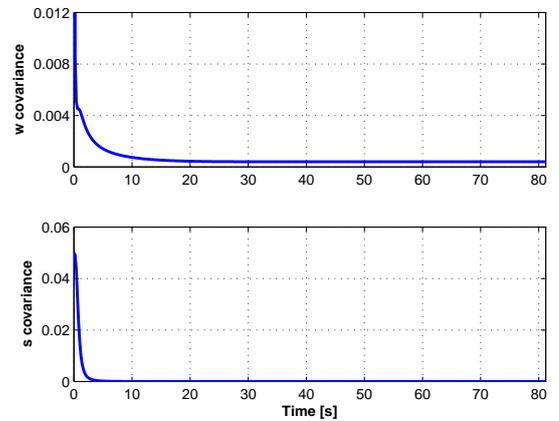


Fig. 14. Covariance evolution for attitude estimators.

#### D. Global Stability of Position Estimator

It is also worth to check the global stability of the localization estimator when it was wrongly initialized on purpose. Thus, to check the stability of the system, some re-localizations were performed considering that the estimated position starts from the different initial positions that are presented in table I.

TABLE I  
INITIAL CONDITIONS OF POSITION STABILITY VALIDATION IN A  
LAWN-MOWER TRAJECTORY

	$x$ position [m]	$y$ position [m]
Robot position	3.6	2.4
Re-localization 1	4.1	2.4
Re-localization 2	4.1	1.9
Re-localization 3	2.1	1.6

The results (figures 15 and 16) show that in spite of the initial large error of the estimated position, the system converges to the correct localization for any different conditions. Analyzing figure 16 it is possible to observe the quickness of the position estimator to stabilize (approximately 5 seconds), what considering the speed of mobile robot of  $0.1 \text{ m} \cdot \text{s}^{-1}$

represents 0.5 m of the travelled distance.

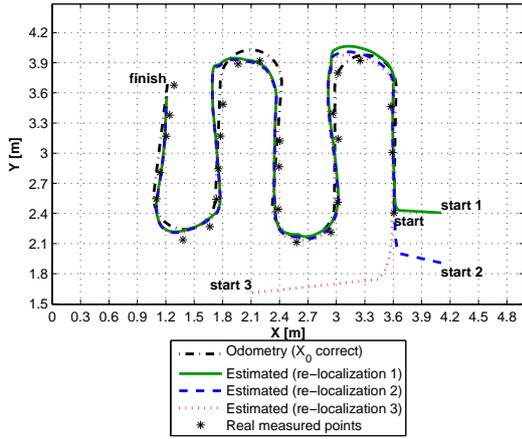


Fig. 15. Results of position stability tests for a Lawn-Mower trajectory (map 2D).

### E. Global Stability of Attitude Estimator

Another important stability question is to assure that the estimator is able to correct the trajectory when the mobile robot starts with a different attitude. This challenge is greater than the position estimation because the robot odometry makes the estimator to describe a different trajectory, what could originate increasing position errors along time.

TABLE II  
INITIAL CONDITIONS OF ATTITUDE STABILITY VALIDATION IN A LAWN-MOWER TRAJECTORY

	attitude [ ° ]
Robot position	90
Re-localization 1	10
Re-localization 2	180
Re-localization 3	270

Thus, considering the initial conditions of table II, the results presented in figure 17 show that the attitude Kalman filter is also fast to estimate the real robot attitude, spending less than 0.5 s to correct it. Once the proposed model for differential mobile robots is a LPV model in function of the angular speed (based in estimated attitude), the quickness to correct the initial wrong attitude allows an estimated position correction in about 6 s.

Moreover, figure 18 shows that only with odometry the mobile robot will be completely lost after starts from a wrong initial direction. However, how it is possible to observe, namely making a zoom at the initial path, the attitude estimator is able to stabilize any wrong initial direction, even considering the worst scenario (in opposite direction).

### F. Global Stability with Wrong Initial Position and Attitude

Finally, a more robustness stability validation test was performed combining the two tests described above. Thus, as initial conditions for the mentioned two tests, the three

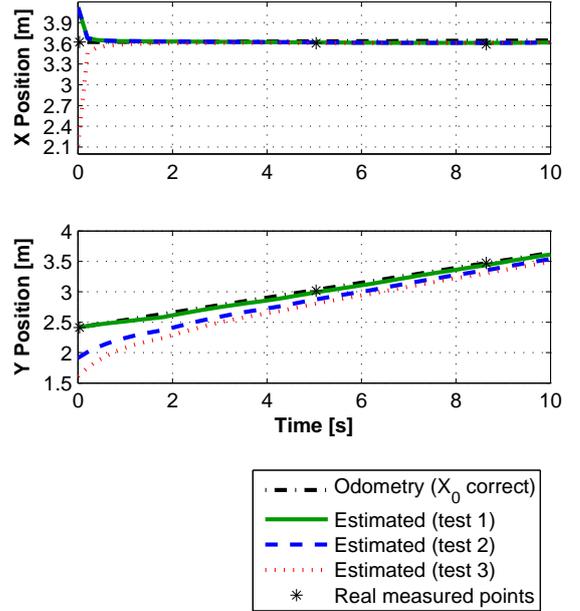


Fig. 16. Results of position stability tests for a Lawn-Mower trajectory (first 10s).

different initial conditions used in the previous tests for the Lawn-Mower trajectory have still been considered but, instead, to respect the correct attitude of the robot, it was applied a very bad initial condition for the attitude estimator (see table III).

TABLE III  
INITIAL CONDITIONS OF POSITION AND ATTITUDE STABILITY VALIDATION IN A LAWN-MOWER TRAJECTORY

	x position [m]	y position [m]	attitude [ ° ]
Robot position	3.6	2.4	90
Re-localization 1	4.1	2.4	0
Re-localization 2	4.1	1.9	315
Re-localization 3	2.1	1.6	225

The results depicted in figure 19 shows that considering the wrong initial conditions, the robot would describe, in an open-loop, the same Lawn-Mower trajectory, but in a different direction. Considering these conditions, the position estimator could diverge from the robot real position, but instead of that, the results are always stable and very fast in their convergence.

### G. Angular Slippage Estimation

Another important aspect to be considered in self-estimation position is the existence of an angular slippage caused by systematic errors, due to the uncertainty in the dimensions of the robot, which greatly increases the error of odometry self-localization. As was presented in the observability study, the model used to estimate the robot attitude

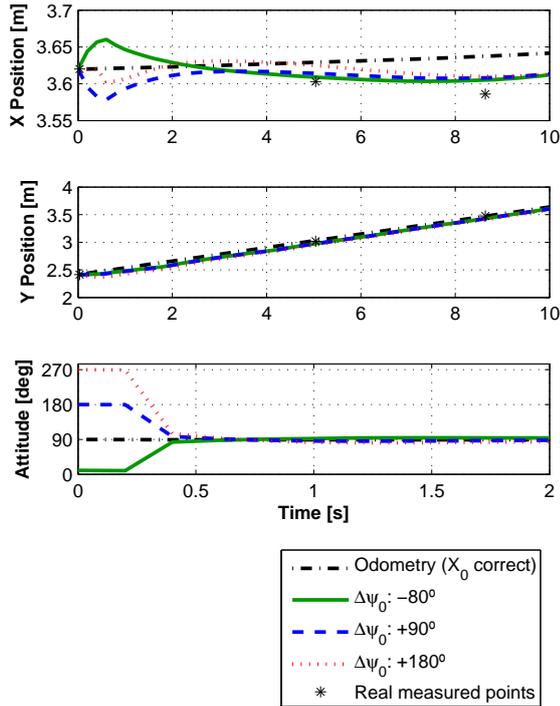


Fig. 17. Results of attitude stability tests for a Lawn-Mower trajectory (first 10s).

is able to estimate this aforementioned angular slippage. To validate the origin and the angular slippage estimation, considering the differential mobile robot presented above, new experimental tests have been conducted making the robot following a similar Lawn-Mower trajectory with a imposed angular slippage of  $1 \text{ deg}\cdot\text{s}^{-1}$ . The results presented in figure 20 shows that the position estimation is not possible with the odometry, once the angular slippage imposed will change the robot real trajectory. However, the proposed self-localization method is able to cope with angular slippages estimating its value in approximately 40 s (see figure 21).

## V. FUTURE WORK

This paper represents the initial step towards an agent based architecture where a large set of mobile robots will be able to cooperate to perform navigation and formation tasks, featuring obstacle avoidance, human interaction and search and rescue activities. The multi-agent system (MAS) to be developed in the near future is described in the remaining of this section.

Clearly, from all the above, it is desired that the mobile robots constitute a MAS, where each of the robots is an intelligent agent [38]. It is desired that each robot to be autonomous, reactive to perceived changes in the environment, pro-active (so as to take initiatives to fulfill its assigned goals), and able to interact with other robots. These are the characteristics that define an intelligent agent. It is also clear that

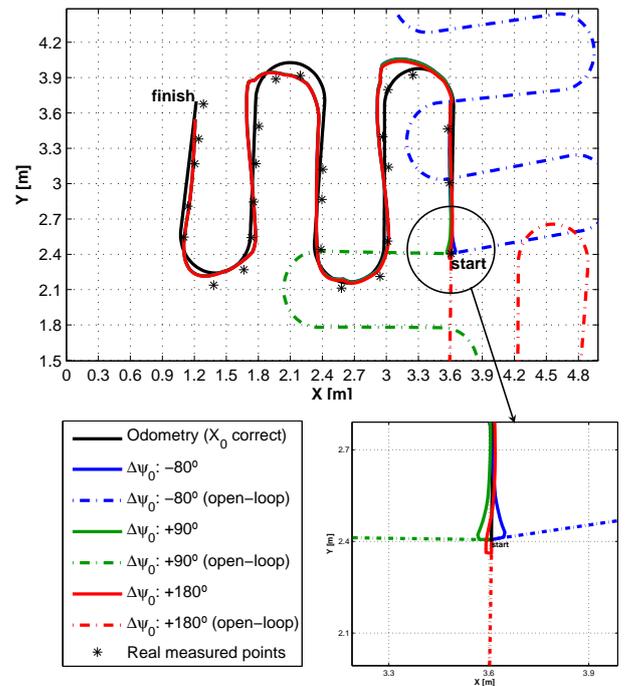


Fig. 18. Results of attitude stability tests for a Lawn-Mower trajectory (map 2D).

- the behavior of the agents is to result both from stimulus–response relations and from a map of the environment from which future actions are decided; i.e. agents are hybrid, inbetween purely reactive and purely cognitive ones;
- agents move in an open environment, i.e. one which is continuous, dynamic, non-deterministic (inasmuch its changes do not solely depend on actions from the agents) and partially inaccessible (as no perfect map is ever got).

Actually multi-robot systems are a classical application of MAS, be it for determining trajectories for different robots avoiding collisions [20], [29], for integrating maps established by different robots taking into account odometry uncertainties [19], or for achieving cooperation of different robots to carry a weight through a door [1], among many other possible examples.

Since several different tasks are to performed by each agent simultaneously, each of the agents corresponding to one mobile robot may be further decomposed in agents for each task, as done e.g. in [30] for map-building from visual data, in [13] for manipulator robots, or in [27] for fault detection and isolation (FDI); or else a conventional parallel computing programming technique may be used (which can often be a better option [38]).

While in this paper, as explained above in section III, only one robot was considered, the architecture schematized in figure 22 can even now be proposed: the tasks within each robot are organized using a hierarchical architecture; communications between agents are also organized hierarchically, with a centralized node responsible for assigning trajectories and keeping an up-to-date map of all regions explored so far; if communications with that center fail, a decentralized

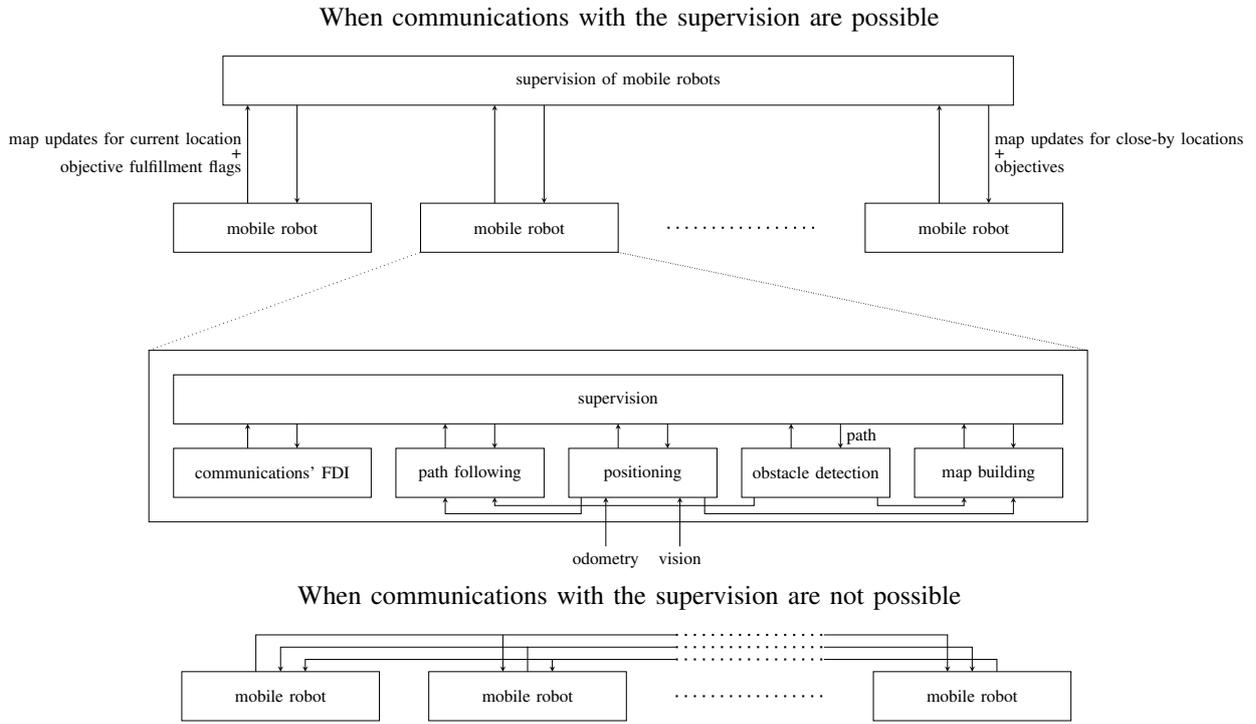


Fig. 22. Multi-agent system to control the mobile robots

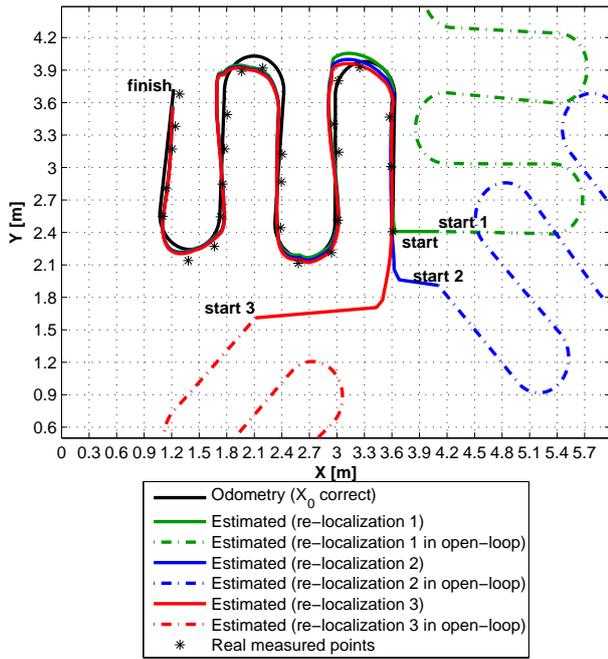


Fig. 19. Results of stability tests considering a wrong initial position and attitude (map 2D).

architecture is used instead for updating the maps, while agents keep their last assigned missions. Actually it might be possible to use always a decentralized architecture, at least for exchanging information on maps; but this option requires far more information exchanges between agents, and thus is less desirable than a hierarchical architecture.

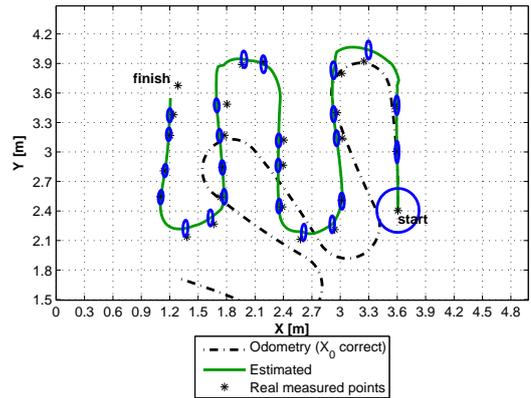


Fig. 20. Results of position estimation with imposed angular slippage (map 2D).

## VI. CONCLUSIONS

A new positioning sensor and a localization system for mobile robots to operate with only onboard sensors in unstructured environments is proposed and experimentally validated. The positioning sensor resorts to PCA, from the images acquired by a camera installed onboard, looking upwards to the ceiling. For the estimation using Kalman filters, a new linear Differential Drive car kinematic model for digital processing was proposed. This model estimates positions and slippages for differential motion guaranteeing a global stability due the properties of the Kalman filter. Several experimental tests was performed using Lawn-Mower trajectory: i) self-localization tests with ground truth

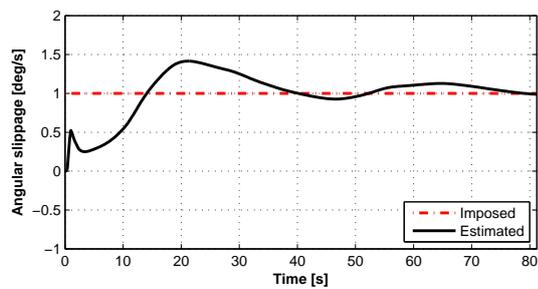


Fig. 21. Results of angular slippage estimation.

validation and Monte Carlo performance study ii) global stability validation for position and attitude estimation and iii) real-time angular slippage estimation. Results shows that the position sensor is very fast to stabilize and guarantees a global stability, even when the estimator starts from wrong positions and attitudes. The proposed method is efficient to be successfully implemented in mobile robots to operate in unstructured environments.

Considering, that the robot used in the experimental validations has a low cost hardware (motors with encoders, webcam, digital compass and a netbook), it is also possible that the proposed model and estimator could be applied in all tasks of Differential Drive car mobile robot that address self-localization as trajectory control and mapping and formations or collaborative work of multi-robots. This paper represents the initial step towards a multi-agent system based architecture where a large set of mobile robots will be able to cooperate to perform navigation and formation tasks, featuring obstacle avoidance, human interaction and search and rescue activities.

## VII. ACKNOWLEDGMENTS

This work was partially supported by Fundação para a Ciência e a Tecnologia, through IDMEC under LAETA.

## REFERENCES

- [1] Mohd Ridzuan Ahmad, Shamsudin H.M. Amin, and Rosbi Mamat. Development of reactive control strategy for multi-agent mobile robotics system. *Jurnal Teknologi*, 35(D):99–114, 2001.
- [2] Matej Artač, Matjaž Jogan, and Aleš Leonardis. Mobile robot localization using an incremental eigenspace model. In *IEEE International Conference on Robotics and Automation*, 2002.
- [3] B. Bacca, J. Salvi, and X. Cufí. Appearance-based mapping and localization for mobile robots using a feature stability histogram. *Robotics and Autonomous Systems*, 59(10):840–857, 2011.
- [4] T. Bailey and H. Durrant-Whyte. Simultaneous localization and mapping (SLAM): part II. *Robotics Automation Magazine, IEEE*, 13(3):108 – 117, sept. 2006.
- [5] W. Baker and R. Clem. Terrain contour matching (tercom) primer. Technical Report ASD-TR-77-61.
- [6] Pedro Batista, Carlos Silvestre, and Paulo Oliveira. Optimal position and velocity navigation filters for autonomous vehicles. *Automatica*, 46(4):767–774, 2010.
- [7] C. Cardeira and J. Sá da Costa. A low cost mobile robot for engineering education. In *Industrial Electronics Society, 2005. IECON 2005. 31st Annual Conference of IEEE*, pages 2162–2167, Raleigh, 2005.
- [8] K. Dahia D. Pham and C. Musso. A kalman-particle kernel filter and its application to terrain navigation. In *Proceedings of the Sixth International Conference of Information Fusion*, page 11721179, Queensland, Australia, july 2003.
- [9] H. Durrant-Whyte and T. Bailey. Simultaneous localization and mapping: part I. *Robotics Automation Magazine, IEEE*, 13(2):99 – 110, june 2006.
- [10] Martin A. Fischler and Robert C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.
- [11] Yasuaki Fukutani, Tomoyuki Takahashi, Masahiro Iwahashi, Tetsuya Kimura, Samsudin Siti Salbiah, and Norrima Binti Mokhtar. Robot vision network based on ceiling map sharing. In *IEEE International Workshop on Advanced Motion Control*, pages 164–169, 2010.
- [12] Arturo Gil, Oscar Martinez Mozos, Monica Ballesta, and Oscar Reinoso. A comparative evaluation of interest point detectors and local descriptors for visual SLAM. *Machine Vision and Applications*, 21(6):905–920, 2010.
- [13] A. Hentout, B. Bouzouia, and Z. Toukal. Multi-agent architecture model for driving mobile manipulator robots. *International Journal of Advanced Robotic Systems*, 5(3):257–268, 2008.
- [14] Marius Hofmeister, Maria Liebsch, and Andreas Zell. Visual self-localization for small mobile robots with weighted gradient orientation histograms. In *International Symposium on Robotics*, pages 87–91, 2009.
- [15] L. Hostetler and R. Andreas. Nonlinear kalman filtering techniques for terrain-aided navigation. *IEEE Transactions on Automatic Control*, AC-28:315–323, March 1983.
- [16] WooYeon Jeong and Kyoung Mu Lee. CV-SLAM: a new ceiling vision-based SLAM technique. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3195–3200, 2005.
- [17] I. Jolliffe. *Principal Component Analysis*. Springer-Verlag, 2002.
- [18] R. Karlsson and F. Gustafsson. Particle filter for underwater terrain navigation. In *IEEE Workshop on Statistical Signal Processing*, page 526529, St. Louis, MO, USA, september 2003.
- [19] Yong-Jae Kim and Jong-Hwan Kim. Online map building evolutionary algorithm for multi-agent mobile robots with odometric uncertainty. In *Congress on Evolutionary Computation*, volume 1, pages 133–138, 2000.
- [20] Fedor A. Kulushev and Alexander A. Bogdanov. Multi-agent optimal path planning for mobile robots in environment with obstacles. In D. Bjørner, M. Broy, and A. Zamulin, editors, *PSI’99, LNCS 1755*, pages 503–510. Springer-Verlag, Berlin, 2000.
- [21] Ben Kröse, Roland Bunschoten, Stephan Ten Hagen, Bas Terwijn, and Nikos Vlassis. Household robots look and learn: environment modeling and localization from an omnidirectional vision system. *IEEE Robotics & Automation Magazine*, 11:45–52, 2004.
- [22] Bor-Woei Kuo, Hsun-Hao Chang, Yung-Chang Chen, and Shi-Yu Huang. A light-and-fast slam algorithm for robots in indoor environments using line segment map. *Journal of Robotics*, 2011:—, 2011.
- [23] I. Loevsky and I. Shimshoni. Reliable and efficient landmark-based localization for mobile robots. *Robotics and Autonomous Systems*, 58(5):520–528, 2010.
- [24] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.
- [25] M. Aldon M. Sistiaga, J. Opperbecke and V. Rigaud. Map based underwater navigation using a multibeam echosounder. In *OCEANS 98 Conference Proceedings*, page 747751, Nice, France, september 1998.
- [26] S. Maeda, Y. Kuno, and Y. Shirai. Active navigation vision based on eigenspace analysis. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 2, pages 1018–1023, 1997.
- [27] Mário J. G. C. Mendes, Bruno M. S. Santos, and José Sá da Costa. Multi-agent platform and toolbox for fault tolerant networked control systems. *Journal of Computers*, 4(4):303–310, 2009.
- [28] Michael Montemerlo, Sebastian Thrun, Daphne Roller, 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*, 2003.
- [29] Mehdi Mouad, Lounis Adouane, Pierre Schmitt, Djamel Khadraoui, and Philippe Martinet. Control architecture for cooperative mobile robots using multi-agent based coordination approach. In *6th National Conference on “Control Architectures of Robots”*, 2011.
- [30] Rafael Muñoz-Salinas, Eugenio Aguirre, Miguel García-Silvente, and Moisés Gómez. A multi-agent system architecture for mobile robot navigation based on fuzzy and visual behaviour. *Robotica*, 23:689–699, 2005.
- [31] I. Nygren and M. Jansson. Robust terrain navigation with the correlation method for high position accuracy. In *OCEANS 2003 Marine Technology and Ocean Science Conference*, volume 3, page 12691277, San Diego, CA, september 2003.
- [32] Paulo Oliveira. MMAE terrain reference navigation for underwater vehicles using PCA. *International Journal of Control*, 80(7):1008–1017, July 2007.

- [33] Z. Yee R. Zelenka and A. Zirkler. Flight test development and evaluation of a kalman filter state estimator for low-altitude flight. In *Second IEEE Conference on Control Applications*, page 783790, Vancouver, Canada, september 1993.
- [34] Davide Scaramuzza, Friedrich Fraundorfer, and Roland Siegwart. Real-time monocular visual odometry for on-road vehicles with 1-point ransac. In *IEEE-ICRA09 International Conference on Robotics and Automation, 2009.*, pages 4293 –4299, may 2009.
- [35] Wolfram Burgard Sebastian Thrun and Dieter Fox. *Probabilistic Robotics*. Intelligent Robotics and Autonomous Agents. MIT Press, 2005.
- [36] C. Siagian and L. Itti. Biologically inspired mobile robot vision localization. *IEEE Transactions on Robotics*, 25(4):861–873, 2009.
- [37] Miroslav Trajković and Mark Hedley. Fast corner detection. *Image and Vision Computing*, 16(2):75–87, 1998.
- [38] M. Wooldridge. *An Introduction to Multiagent Systems*. John Wiley & Sons, Chichester, 2002.
- [39] De Xu, Liwei Han, Min Tan, and You Fu Li. Ceiling-based visual positioning for an indoor mobile robot with monocular vision. *IEEE Transactions on Industrial Electronics*, 56(5):1617–1628, 2009.