Single-Camera EKF-*v*SLAM

ML. Benmessaoud, A. Lamrani, K. Nemra, and AK. Souici

***Abstract*—**This paper presents an Extended Kaman Filter implementation of a single-camera Visual Simultaneous Localization and Mapping algorithm, a novel algorithm for simultaneous localization and mapping problem widely studied in mobile robotics field. The algorithm is vision and odometry-based, The odometry data is incremental, and therefore it will accumulate error over time, since the robot may slip or may be lifted, consequently if the odometry is used alone we can not accurately estimate the robot position, in this paper we show that a combination of odometry and visual landmark via the extended Kalman filter can improve the robot position estimate. We use a Pioneer II robot and motorized pan tilt camera models to implement the algorithm.

***Keywords*—**Mobile Robot, Navigation, *v*SLAM, EKF, monocular.

I. Introduction

SIMULTANEOUS Localization And Mapping (SLAM) is one of the most fundamental, yet most challenging problems in mobile robotics. To achieve full autonomy a robot must possess the ability to explore its environment without user intervention, build a reliable map, and localize itself in the map. In particular, if *global positioning sensor* (GPS) data and external beacons are unavailable, the robot must somehow, by itself, determine what are appropriate reference points, on which to use as reference to localize itself**.**

Successful implementations of SLAM have generally been achieved with laser, sonar or stereo vision range sensors and built maps for controlled robots moving in 2D or 3D real environment [1], [2], [3]. Solving the SLAM problem with vision as the only external sensor is now the goal of much of the effort in the area. Monocular vision is especially interesting as it offers a highly affordable solution in terms hardware. Recent research has proved that real-time *Visual Simultaneous Localization and Mapping* (*v*SLAM) can be achieved using monocular vision as sensory input and using only weak motion modeling [5], [6], indicating not only that vision will become increasingly important as a cheap, compact and flexible tool for robot navigation but that visual SLAM will be able to play a role in other domains in which automatic localization is required.

Manuscript submitted March, 2008

Mohamed Lamine Benmessaoud is with Military Polytechnic School BP 17 M1 Bordj Elbahri, Algiers Algeria, 16111.

Aissa Lamrani is with Military Polytechnic School BP 17 M1 Bordj Elbahri, Algiers 16111, Algeria.

Karim Nemra is with Cranfield University UK.

Souici abd elkarim is with the robotic laboratory, Military Polytechnic School, BP 17 M1 Bordj Elbahri, Algiers 16111, Algeria (e-mail: aks752005@yahoo.fr). And he is preparing doctorat thesis in collaboration with LAAS/CNRS, France (e-mail: aesouici@laas.fr).

In this work we present an extended Kaman Filtere (EKF) implementation of *v*SLAM algorithm using motorized pan tilt camera mounted on Pioneer II robot. We have developed a complete model of both robot and camera in their environment (global frame).

The paper is organized as follows. Next section states the problem solved by *v*SLAM. The third section gives the complete model of the robot and the motorize pan tilt camera in a global frame. Section IV gives an overview of the EKF-*v*SLAM algorithm and describes its execution procedure. Section V consists of simulations that illustrate the performance of EKF-*v*SLAM. Finally, in Section VI we draw some conclusions, some final remarks and perspectives.

II. Problem Formulation

Consider a mobile robot that must localize itself in a previously unknown environment. The objective is to choose a sensor configuration and an algorithm to process the sensor data, which accurately and robustly accomplish the localization in real-world environments.

Assume it is not accepted to alter the environment by installing beacons or other external equipment. That is, the design choices only apply to the robot itself, and the robot must determine its location based only on data collected by the selected onboard sensors.

Also, realize that since the environment is unknown, the robots must also map the environment. Now, suppose the mobile robot is equipped with an odometry sensor providing velocity data, and a single camera providing images of the environment. The odometry sensor may consist of standard wheel encoders attached to the driving wheels of a differential drive system.

The objective of the *v*SLAM system is to fuse image and odometry data in a way that enables robust map-building and localization. Being "robust" is key since the sensor data acquired from the mobile robot (odometry and images) contains plenty of difficult-to-model noise. The odometry data is incremental, and therefore it will accumulate error over time. Understand first of all that the odometry sensor can never be perfectly calibrated. But, since the robot may slip or may be lifted, the odometry is also exposed to discrete events of dramatic errors.

III. Robot Camera Models

A. Cinematic Model

The configuration of a rigid mobile robot is commonly described by six variables, its three-dimensional cartesian coordinates and its three Euler angles (roll, pitch, yaw), relative to an external coordinate frame. The material presented in this paper is largely restricted to mobile robots

operating in planar environments, whose kinematic state, or pose, is summarized by three variables. This is illustrated in Fig. 1. The robot's pose comprises its two-dimensional planar coordinates relative to an external coordinate frame, along with its angular orientation. The following vector describes the pose of the robot:

$$R = (x \quad y \quad \theta)^T \tag{1}$$

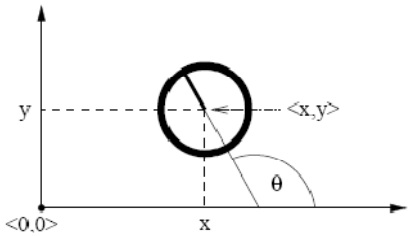

Fig. 1 Robot configuration

The non-holonomic constraint is given by the following equation:

$$\dot{x}\sin(\theta) - \dot{y}\cos(\theta) = 0 \tag{2}$$

There is no translation following y_r axis of the robot local frame (Fig. 1), after sampling we obtain:

$$\begin{bmatrix} x(k+1) \\ y(k+1) \\ \theta(k+1) \end{bmatrix} = \begin{bmatrix} x(k) + v_k \cos\theta(k)T \\ y(k) + v_k \sin\theta(k)T \\ \theta(k) + \omega_k T \end{bmatrix} \tag{3}$$

Where T is the sampling period, v_k and ω_k, are the translation and the rotation velocity of the robot respectively at sample k. We will denote the measured velocity vector at sample k by $u_k = (v_k \quad w_k)^T$, therefore equation (3) can be noted:

$$R_{k+1} = c(R_k, u_k) \tag{4}$$

The above model states the cinematic for an ideal, noise-free robot. **In reality, robot motion is subject to noise, slip or lift**. The actual velocities differ from the measured ones by odometry sensor [7]. We will model this difference by a random variable with finite variance. More precisely, let us assume the actual velocities are given by:

$$\breve{u}_k = \begin{pmatrix} \breve{v}_k \\ \breve{w}_k \end{pmatrix} = \begin{pmatrix} v_k \\ w_k \end{pmatrix} + \begin{pmatrix} \varepsilon_v \\ \varepsilon_w \end{pmatrix} \tag{5}$$

The true velocity equals the measured velocity plus some small, additive error (noise) [7]. The Fig. 2 shows a simulation of the real robot trajectory and the trajectory measured by odometry sensor.

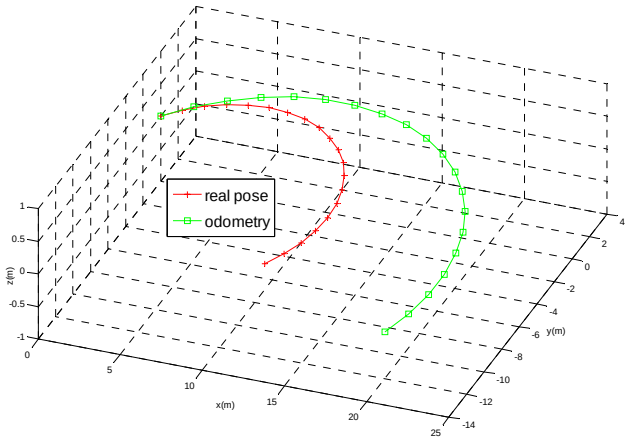

Fig. 2 Robot motion

B. *Observation Model*

The observation model is a key point for SLAM implementation [5] [6] [7], as reported in the problem formulation, the robot must choose appropriate reference or landmark in the environment to localize itself; these landmark must be stable and invariant. Since we use image as primary sensor, these landmark must be extracted, like corner, edges or regions, in the literature we can found a several technique to extract robust landmark from image, for example Harris corner detector algorithm is commonly used [6] (Fig. 3).

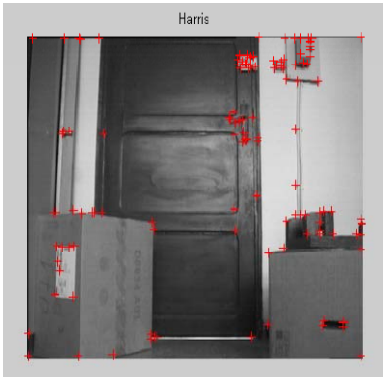

Fig. 3 Harris corner detection

In this paragraph we represent the robot and the camera in their environment or global frame. The camera is mounted on the front of the robot (Fig. 4), and it is motorized in pan and tilt axes (Fig. 5).

Fig. 4 Pioneer II Robot and PTZ camera

We will project the landmarks represented in the global frame, on the image frame, through projections between successive frames (global, robot, Pan, Tilt, Camera, and image) (Fig. 6).

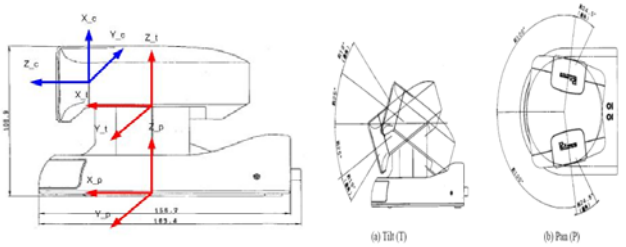

Fig. 5 The PTZ camera and pan /tilt rotation

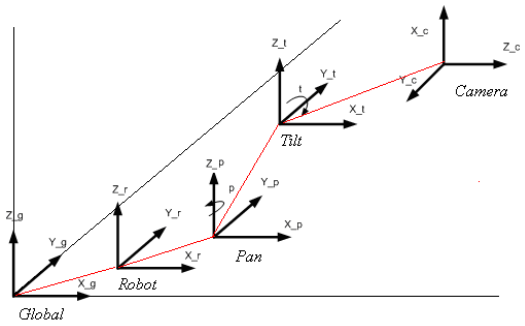

Fig. 6 Successive frames

The successive projections matrixes are given as follow:

a) *Global to Robot Projection*

$$M_{GR}=\begin{bmatrix}\cos(\theta) & -\sin(\theta) & 0 & x\\ \sin(\theta) & \cos(\theta) & 0 & y\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\end{bmatrix}$$

b) Robot to Pan Projection

$$M_{RP}=\begin{bmatrix}\cos(P) & \sin(P) & 0 & x_{PR}\\ -\sin(P) & \cos(P) & 0 & 0\\ 0 & 0 & 1 & z_{PR}\\ 0 & 0 & 0 & 1\end{bmatrix}$$

Were P is the pan angle and x_{pr}, z_{pr} are the translations between robot and pan frames (Fig. 5).

c) Pan to tilt projection:

$$M_{PT}=\begin{bmatrix}\cos(T) & 0 & -\sin(T) & x_{TP}\\ 0 & 1 & 0 & 0\\ \sin(T) & 0 & \cos(T) & z_{TP}\\ 0 & 0 & 0 & 1\end{bmatrix}$$

Were T is the tilt angle and x_{TP}, z_{TP} are the translations between pan and tilt frames (Fig. 5).

d) Tilt to Camera Projection

$$M_{TC}=\begin{bmatrix}0 & 0 & -1 & x_{CT}\\ 0 & -1 & 0 & y_{CT}\\ 1 & 0 & 0 & z_{CT}\\ 0 & 0 & 0 & 1\end{bmatrix}$$

x_{CT}, y_{CT}, z_{CT} are the translations between camera and tilt frames (Fig. 5).

The projection from global coordinate to camera coordinate can be found as follow:

$$M_{CG}=(M_{GR}M_{RP}M_{PT}M_{TC})^{-1} \tag{6}$$

Let us not the landmarks global and camera coordinate respectively by:

$$L^g=\begin{pmatrix}L_x^g & L_y^g & L_z^g\end{pmatrix}^T,\ L^c=\begin{pmatrix}L_x^c & L_y^c & L_z^c\end{pmatrix}^T$$

Thus we can write:

$$\begin{pmatrix}L_x^c\\ L_y^c\\ L_z^c\end{pmatrix}=M_{CG}\begin{pmatrix}L_x^g\\ L_y^g\\ L_z^g\end{pmatrix} \tag{7}$$

e) Camera to Image projection:

Let us note the image coordinate by: $I=\begin{pmatrix}u & v\end{pmatrix}^T$

$$I=\begin{cases}u=\alpha_u\dfrac{L_x^c}{L_z^c}+u_0\\ v=\alpha_v\dfrac{L_y^c}{L_z^c}+v_0\end{cases} \tag{8}$$

Were α_u, α_v, u_0 and v_0 are the intrinsic characteristics of the camera.

Replacing equations (6) and (7) in equation (8) we obtain:

$$\begin{cases}u=\alpha_u\left(\dfrac{M_{11}L_x^g+M_{12}L_y^g+M_{13}L_z^g+M_{14}}{M_{31}L_x^g+M_{32}L_y^g+M_{33}L_z^g+M_{34}}\right)+u_0\\ v=\alpha_v\left(\dfrac{M_{21}L_x^g+M_{22}L_y^g+M_{23}L_z^g+M_{24}}{M_{31}L_x^g+M_{32}L_y^g+M_{33}L_z^g+M_{34}}\right)+v_0\end{cases} \tag{9}$$

M_{ij} are parameters depending on the robot configuration, Pan and Tilt angles.

Finally, we obtain a model representing the image coordinate of a landmark according to its 3D coordinates and the robot configuration. This model is called the direct observation model and will be noted:

$$I=h(R,L^g) \tag{10}$$

In reality the observation is subject to noise. The real measurement model is given in the following model:

$$\breve{I}=h(R,L^g)+\begin{pmatrix}\alpha_u\\ \alpha_v\end{pmatrix} \tag{11}$$

C. Inverse Observation Model

Regarding the dynamic nature of the SLAM algorithm, new observed landmark must be initialized prior to be added to the state vector [7]. The initialization process is in fact the best estimation of the new landmark position, and it is a fundamental point to SLAM implementation.

The observation model stated in (10) gives two equations for three dimension variable L^g. Using one image will not solve the system, so we need two images for the same landmark to solve the system.

Let $I_1=(u_1,v_1)^T$, $I_2=(u_2,v_2)^T$ be the images coordinates of a landmark, respectively in the first and second image, taken from two different robot pose respectively R, R'. If these two coordinate are associated correctly then we can write the following equation system.

$$\begin{cases} u_1=\alpha_u\left(\dfrac{{}_1M_{11}L_x^g+{}_1M_{12}L_y^g+{}_1M_{13}L_z^g+{}_1M_{14}}{{}_1M_{31}L_x^g+{}_1M_{32}L_y^g+{}_1M_{33}L_z^g+{}_1M_{34}}\right)+u_0 \\ v_1=\alpha_v\left(\dfrac{{}_1M_{21}L_x^g+{}_1M_{22}L_y^g+{}_1M_{23}L_z^g+{}_1M_{24}}{{}_1M_{31}L_x^g+{}_1M_{32}L_y^g+{}_1M_{33}L_z^g+{}_1M_{34}}\right)+v_0 \\ u_2=\alpha_u\left(\dfrac{{}_2M_{11}L_x^g+{}_2M_{12}L_y^g+{}_2M_{13}L_z^g+{}_2M_{14}}{{}_2M_{31}L_x^g+{}_2M_{32}L_y^g+{}_2M_{33}L_z^g+{}_2M_{34}}\right)+u_0 \\ v_2=\alpha_v\left(\dfrac{{}_2M_{21}L_x^g+{}_2M_{22}L_y^g+{}_2M_{23}L_z^g+{}_2M_{24}}{{}_2M_{31}L_x^g+{}_2M_{32}L_y^g+{}_2M_{33}L_z^g+{}_2M_{34}}\right)+v_0 \end{cases} \quad (12)$$

The above equation system can be transformed to the following linear system:

$$E.L^g=b \quad (13)$$

Were:

$$E=\begin{bmatrix} \frac{(u_1-u_0)}{\alpha_u}{}_1M_{31}-{}_1M_{11} & \frac{(u_1-u_0)}{\alpha_u}{}_1M_{32}-{}_1M_{12} & \frac{(u_1-u_0)}{\alpha_u}{}_1M_{33}-{}_1M_{13} \\ \frac{(v_1-v_0)}{\alpha_v}{}_1M_{31}-{}_1M_{21} & \frac{(v_1-v_0)}{\alpha_v}{}_1M_{32}-{}_1M_{22} & \frac{(v_1-v_0)}{\alpha_v}{}_1M_{33}-{}_1M_{23} \\ \frac{(u_2-u_0)}{\alpha_u}{}_2M_{31}-{}_2M_{11} & \frac{(u_2-u_0)}{\alpha_u}{}_2M_{32}-{}_2M_{12} & \frac{(u_2-u_0)}{\alpha_u}{}_2M_{33}-{}_2M_{13} \\ \frac{(v_2-v_0)}{\alpha_v}{}_2M_{31}-{}_2M_{21} & \frac{(v_2-v_0)}{\alpha_v}{}_2M_{32}-{}_2M_{22} & \frac{(v_2-v_0)}{\alpha_v}{}_2M_{33}-{}_2M_{23} \end{bmatrix}$$

$$b=\begin{bmatrix} {}_1M_{14}-\frac{(u_1-u_0)}{\alpha_u}{}_1M_{34} \\ {}_1M_{24}-\frac{(v_1-v_0)}{\alpha_v}{}_1M_{34} \\ {}_2M_{14}-\frac{(u_2-u_0)}{\alpha_u}{}_2M_{34} \\ {}_2M_{24}-\frac{(v_2-v_0)}{\alpha_v}{}_2M_{34} \end{bmatrix}$$

The 3D coordinates of a new landmark are initialized by solving (13) using Least Square technique. We obtain the best estimation of new landmark position as follow:

$$L^g=(E^TE)^{-1}.E^T.b \quad (14)$$

We note:

$$L^g=h^{-1}(R,I_1,R',I_2) \quad (15)$$

The following figure shows a simulation environment developed using models (3) (10) and (15).

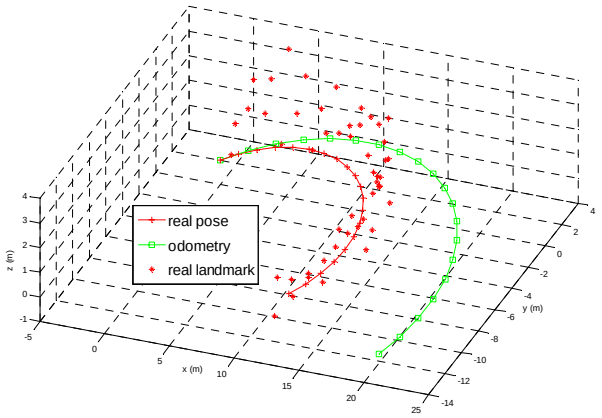

Fig.8. simulation environment

IV. Overview of the EKF-*v*SLAM Algorithm

The EKF-*v*SLAM algorithm applies the EKF to the SLAM problem using vision as sensor. In doing so, EKF-*v*SLAM is subject to the **Gaussian noise** assumption as any EKF algorithm; EKF-*v*SLAM makes a Gaussian noise assumption for the robot motion and the observation model. In addition to estimating the robot pose *R*, the EKF-*v*SLAM algorithm also estimates the coordinates of all landmarks encountered along the way. This makes it necessary to include the landmark coordinates into the state vector. For convenience, let us call the state vector comprising robot pose and the map the *combined state vector*, and denote this vector y_k. The combined vector is given by:

$$y_k=\begin{pmatrix} R_k \\ \underline{L}^{g,k} \end{pmatrix}$$

Were: $R_k=(x_k \quad y_k \quad \theta_k)^T$ is the robot pose at time *k*, given by cinematic model (3)(4), $\underline{L}^{g,k}=\left(L_1^{g,k} \quad L_2^{g,k} \quad ..... \quad L_N^{g,k}\right)^T$ is the map vector containing the entire landmark coordinate at time *k*, *N* number of landmark, and $L_i^{g,k}=\left(L_{i,x}^{g,k} \quad L_{i,y}^{g,k} \quad L_{i,z}^{g,k}\right)^T$ is the i_{th} landmark in the map vector at time *k*.

As the robot moves, the state vector changes according to the standard noise-free cinematic model (3)(4). In SLAM, this motion model is extended to the augmented state vector:

$$y_{k+1}=g(y_k,u_k)=\begin{pmatrix} c(R_k,u_k) \\ \underline{L}^{g,k} \end{pmatrix} \quad (16)$$

Only the first three elements are updated. Landmarks supposed fix, remain where they are.

As usual in EKFs, the motion function *g* is approximated using a first degree Taylor expansion:

$$g(y_k,u_k)=g(\bar{y}_k,u_k)+G_{y,k}(y_k-\bar{y}_k)$$

Were $G_{y,k}(\bar{y}_k,u_k)$ is the Jacobian of *g* according to y_k, and $\bar{y}_k$ is the mean of y_k.

The velocity errors measured by the odometry must be propagated in to state vector, let us note $G_{u,k}(\bar{y}_k,u_k)$ the Jacobian of *g* according to u_k.

The measurement model is deduced from (10), we are given the following measurement model:

$$I_{k,j} = h(y_k) = h(y_{k,i}) = h(R_k, L_i^{g,k}) \quad (17)$$

Were *j* is the index of an individual landmark observation in I_k, and *i* is the index of the observed landmark at time *k*.

The observation function *h* is approximated using a first degree Taylor expansion:

$$h(y_{k,i}) = h(\bar{y}_{k,i}) + H_k^j (y_{k,i} - \bar{y}_{k,i})$$

Here H_k^j is the derivative of *h* with respect to the full state vector y_k. Since *h* depends only on two elements of that state vector, the robot pose R_k and the location of the i_{th} landmark $L_i^{g,k}$, we can write H_k^j as follow:

$$H_k^j = h_k^j F_{y,i}$$

Here h_k^j is the Jacobian of *h* at $\bar{y}_k$, calculated with respect to the state variables R_k and $L_i^{g,k}$.

$$F_{y,i} = \begin{bmatrix} 1 & 0 & 0 & 0\cdots\cdots 0 & 0 & 0 & 0 & 0\cdots\cdots 0 \\ 0 & 1 & 0 & 0\cdots\cdots 0 & 0 & 0 & 0 & 0\cdots\cdots 0 \\ 0 & 0 & 1 & 0\cdots\cdots 0 & 0 & 0 & 0 & 0\cdots\cdots 0 \\ 0 & 0 & 0 & 0\cdots\cdots 0 & 1 & 0 & 0 & 0\cdots\cdots 0 \\ 0 & 0 & 0 & 0\cdots\cdots 0 & 0 & 1 & 0 & 0\cdots\cdots 0 \\ 0 & 0 & 0 & \underbrace{0\cdots\cdots 0}_{3i-3} & 0 & 0 & 1 & \underbrace{0\cdots\cdots 0}_{3N-3i} \end{bmatrix}$$

A. Description of the EKF-vSLAM Execution Procedure

The execution procedure of the EKF-*v*SLAM can be divided into four major steps:

1-The initialization

In SLAM, the initial pose is taken to be to origin of the coordinate system. On the other hand the camera is motorized so we can use different pan/tilt angles to acquire different measures, applying the inverse observation model (15) we can initialize the first landmarks. As a result we obtain the first state vector y_0.

$$\bar{y}_0 = \begin{pmatrix} \bar{R}_0 & \bar{L}_1^{g,0} & \bar{L}_2^{g,0} & ..... & \bar{L}_{N_0}^{g,0} \end{pmatrix}^T$$

$$\Sigma_0 = \begin{bmatrix} 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & \sigma_{a1}^2 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & \sigma_{aN_0}^2 \end{bmatrix}$$

Were $\bar{R}_0 = \begin{pmatrix} 0 & 0 & 0 \end{pmatrix}^T$ and N_0 is the number of initial landmarks.

2-The prediction

- $\hat{y}_{k+1} = g(\bar{y}_k, u_k)$
- $\hat{\Sigma}_{k+1} = G_{y,k+1} \Sigma_k G_{y,k+1}^T + G_{u,k+1} \breve{N}_u G_{u,k+1}^T$

3-The update

- ***For all extracted measures*** $I_{k,j}$
- $\hat{I}_k = h(\hat{y}_{k+1})$ ***% measures prediction***
- ***Find*** $\hat{I}_{k,i}$ ***which correspond to*** $I_{k,j}$
- ***If correspondence is found***
  - $H_k^j = h_k^j F_{y,i}$
  - $S_{k+1}^j = H_k^j \hat{\Sigma}_{k+1} H_k^{jT} + \breve{N}_m$,
  - $Ka_{k,j} = \hat{\Sigma}_{k+1} (H_k^j)^T (S_k^j)^{-1}$
  - $\hat{y}_{k+1} = \hat{y}_{k+1} + Ka_{k,j} (I_{k,j} - \hat{I}_{k,i})$
  - $\hat{\Sigma}_{k+1} = (I - Ka_{k,j} H_k^j) \hat{\Sigma}_{k+1}$
- ***Else***
  - ***Save*** $I_{k,j}$ ***in a buffer*** B_I.
- ***End if.***
- ***End for.***
- $\bar{y}_{k+1} = \hat{y}_{k+1}$, $\Sigma_{k+1} = \hat{\Sigma}_{k+1}$ ***% update***
- ***From*** $\bar{y}_{k+1}$, ***add*** $\bar{R}_{k+1}$ ***in the buffer*** B_I, $\bar{R}_{k+1}$ ***is the best robot position estimate in which*** $I_{k,j}$ ***was saved.***

4- The map management

- ***Look for correspondence between measures saved in*** B_I
- ***If correspondence is found***
  - $N_k = N_k + 1$ ***% add the new landmark***
  - ***Use equation (15) to initialize the new landmark*** $L_{N_k}^{g,k}$
  - ***add*** $L_{N_k}^{g,k}$ ***to the state vector***
  - ***approximate equation (15) to initialize the covariance of landmark*** $L_{N_k}^{g,k}$
  - ***add the new covariance to the state covariance***
- ***End if***
- $N_{k+1} = N_k$ ***% number of landmark at time k+1***

V. Simulation and Results

To illustrate the performance of EKF-*v*SLAM, we present the following results. First, consider (Fig. 9), which shows the result of *v*SLAM after the robot has travelled along the real trajectory. The *v*SLAM algorithm builds a map consisting of landmarks, which are marked as circles in the figure. The green path (odometry *only*) is obviously incorrect (Fig. 10) (Fig. 11). The *v*SLAM corrected path, on the other hand, is consistently following the reference path. The *v*SLAM path, which uses a combination of visual measurements and odometry, provides more accurate position determination for the robot (Fig. 10) (Fig. 12).

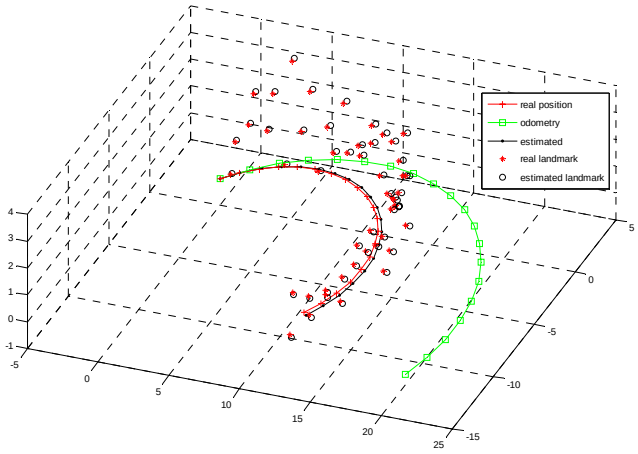

Fig. 9 EKF-*v*SLAM implementation

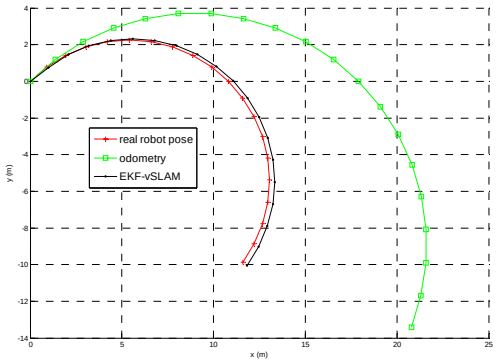

Fig. 10 EKF-*v*SLAM and odometry path

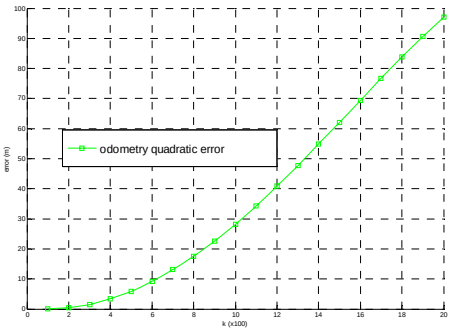

Fig. 11 Odometry quadratic error

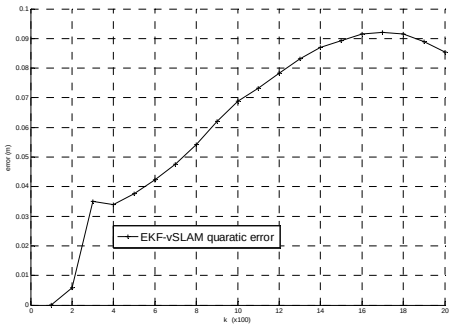

Fig. 12 EKF-*v*SLAM quadratic error

VI. Conclusion

In this work we have developed a complete observation model of a motorized PTZ camera and a cinematic model of Pioneer II robot, and we have implemented monocular *v*SLAM in simulation, which is a novel vision and odometry-based SLAM algorithm that enables low-cost and robust navigation in real environments. The EKF-*v*SLAM algorithm does not require any initial map, and the algorithm is typically good at correcting for slippage and odometry drift.

A problem with the current implementation of *v*SLAM is the state vector size, which in large environments may become prohibitively large, and the algorithm requires memory that is quadratic in *N*, the number of landmarks in the map. Its update time is also quadratic in *N*. The quadratic update complexity stems from the matrix multiplications that take place at various locations in the EKF.

Based on the previous encountered problems, we suggest for future work a solution to maintain a constant number of landmarks in the state vector.

References

[1] S. Thrun, W. Burgard, and D. Fox. «A real-time algorithm for mobile robot mapping with applications to multi-robot and 3D mapping». In: Proceedings of the IEEE International Conference on Robotics and Automation, 2000.

[2] M. Newman, J. J. Leonard, J. Neira, and J. Tardos. «Explore and return: Experimental validation of real time concurrent mapping and localization». In: Proceedings of the IEEE Inter-national Conference on Robotics and Automation, 2002.

[3] J. Davison « 3D Simultaneous Localization and Map-Building Using Active Vision for a Robot Moving on Undulating Terrain ». In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition , 2001.

[4] J. Davison «Real-Time Simultaneous Localization and Mapping with a Single Camera». In: Proceedings of the Ninth IEEE International Conference on Computer Vision 2003.

[5] M. Tomono. «3-D Localization and Mapping Using a Single Camera Based on Structure-from-Motion with Automatic Baseline Selection». In: Proceeding of the IEEE International Conference on Robotics and Automation, 2005.

[6] C. Harris and M.J Stephens. «A combined corner and edge detector». In: Alvey Vision Conference, Pages 147-152, 1988.

[7] S. Thrun, W. Burgard, and D. Fox. «Probabilistic Robotics». MIT Press 2005.