



HAL
open science

Exploiting the Information at the Loop Closure in SLAM

Agostino Martinelli, Roland Y. Siegwart

► **To cite this version:**

Agostino Martinelli, Roland Y. Siegwart. Exploiting the Information at the Loop Closure in SLAM. ICRA, 2007, Rome, Italy. inria-00359936

HAL Id: inria-00359936

<https://inria.hal.science/inria-00359936v1>

Submitted on 9 Feb 2009

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Exploiting the Information at the Loop Closure in SLAM

A. Martinelli and R. Siegwart

Abstract—This paper presents two methods able to exploit the information at the loop closure in the *SLAM* problem. Both methods have three fundamental advantages. The first one is that to apply the loop closure constraint they do not require to compute any correlation among the features which are not observed simultaneously. The second advantage is that the loop closure constraint can be applied only once at the end (even after more than one loop) in a single step with low computational complexity. Hence, the computational complexity during the robot exploration is independent of the number of features. Finally, the third advantage is that the linearization does not affect the estimation process. This especially holds for the second method, which is based on the Levenberg-Marquardt algorithm. The first method is based on the Extended Kalman Filter. Simulations show that these approaches significantly outperform the conventional *EKF* based *SLAM* both in the computational cost and in the map precision.

I. INTRODUCTION

Simultaneous Localization and Mapping (SLAM) requires a mobile robot to autonomously explore the environment with its on-board sensors, gain knowledge about it, interpret the scene, build an appropriate map and localize itself relative to this map. Many approaches have been proposed both in the framework of metric and topological navigation. A very successful metric method is the stochastic map [19] where early experiments [4] [11] have shown the quality of fully metric SLAM.

Currently, the SLAM has two contrasting problems to be solved, which are often faced with a trade-off:

- The map precision;
- The computational requirement for real-time/real-world implementation

Dissanayake et al. [7], proved the convergence of an algorithm based on the Kalman filter theoretically. However, the proof is based on the strong hypothesis of a linear observation. Julier and Uhlmann [10] and Castellanos et al. [3] proved that the conventional *EKF* based *SLAM* (from now on *EKF-SLAM*) yields an inconsistent map (in particular, in [10] was shown that this happens even for the special case of a stationary vehicle with no process noise). The map inconsistency arises from the linearization introduced by the Extended Kalman Filter (*EKF*) as clearly pointed out by Castellanos et al. [3]. Indeed, this approximation only holds if the difference between the estimated state and the ground truth is small. Now, in any map representation, the corresponding vehicle location will drift (if no loop is

closed). This is a consequence of the fact that the absolute location is derived from a composition of many relative measurements. Therefore, when the drift is large enough, the linearization is not a possible approximation. To solve this big inconvenient there are two options: adopting an optimal filter which accounts the non linearity (e.g. the one introduced by Germani, Manes and Palumbo [8]); or maintaining the estimation process local.

The first option has the inconvenience that the computational complexity will explode. This is an important problem even by adopting the simple *EKF* (i.e. the *EKF-SLAM*) where it is required to update a full covariance matrix resulting in a complexity which scales quadratically with the number of features. To reduce the computational burden, Csorba, Uhlmann and Durrant-Whyte [5] and Deans and Hebert [6] introduced a relative map based on quantities invariant to the robot pose (i.e. to shift and rotation). They estimate the distance between two features, which is shift and rotation invariant. However, their algorithms do not consider the dependency among the distances. In particular, they did not consider this dependency to gain the huge amount of information coming when a loop is closed. Newman introduced a relative map and he used two filters in the estimation, called the relative map filter and the geometric projection filter ([15] and [16]). The second one provides a mean to produce a geometrically consistent map from the relative map, by solving a set of linear constraints. Both filters are based on the Kalman Filter. However, the elements used in this approach are invariant for shift only, not for rotation and therefore the estimation process cannot be local (it becomes local only if the robot orientation is a priori known, as assumed in [15]). The approach adopted in [14], [17] is to take invariant elements for both shift and rotation and to perform the estimation through a Kalman Filter. In [14], [13] the equations of this filter are provided. They can be adopted to estimate the invariants among any kind of features (e.g. points, corners). In [17] the dependency among these invariant elements is considered to improve the precision. However, it is not exploited to gain the huge amount of information when a loop is closed.

In this paper we introduce two simple and very powerful methods able to exploit the information contained in the dependencies among the elements belonging to the relative state, which is shift and rotation invariant. In particular, the dependencies created by closing a loop are fully considered. The proposed methods have three key advantages. The first one is that to apply the loop closure constraint it is not required to compute the correlations among the elements of the estimated state which refer to features not observed

A. Martinelli is with Autonomous System Laboratory, ETHZ, Zurich, Switzerland and with Inria Rhône Alpes, St Ismier Cedex, France agostino.martinelli@ieee.org

R. Siegwart is with Autonomous System Laboratory, ETHZ, Zurich, Switzerland roland.siegwart@ethz.ch

simultaneously. Hence, the computational complexity during the robot exploration is independent of the number of features. The second advantage is that the loop closure constraint can be applied only once at the end (even after more than one loop) in a single step, without losing information. Finally, the third advantage is that the linearization does not affect the estimation process. This especially holds for the second method, which is based on the Levenberg-Marquardt algorithm (*LMA*). The first method is based on the *EKF*. Therefore, the final estimation accuracy is significantly better than in the *EKF-SLAM*. Indeed, the *EKF-SLAM* needs the correlations among the features in order to exploit the information at the loop closure. Their computation not only is very expensive but also approximated since affected by the linearization and by an approximated knowledge of the sensor error model. In contrast, the proposed methods exploit geometrical constraints directly and simply derived from the structure of the environment. In the next section we quickly remind some basic concepts in relative mapping (more details can be found in [13], [14]). The methods are introduced in section III. In section IV simulations show that both methods significantly outperform the *EKF-SLAM* both for the computational cost and for the map precision. As expected, the method based on the *LMA* is able to correct the map even when the map precision before the loop closure is very low. Conclusions are presented in section V.

II. BASIC CONCEPTS IN RELATIVE MAPPING

In the relative map approach to *SLAM* it is estimated a state I , which contains relative quantities among the features invariant under shift and rotation (e.g. distances, angles). Once the relative map has been estimated and the absolute location of a set of features is known (the seeding features, whose location could be obtained through the first observations) it is possible to build the absolute map. Therefore, the entire method contains two algorithms. The former estimates the relative map, the latter builds the absolute map. The computational complexity required to build the absolute map from the relative state scales linearly with the number of features ($O(N)$). We remark that it is not necessary to compute the absolute map at each step starting from the seeding features since the relative map estimation is carried out locally. Therefore, the absolute map (if needed) can be in general updated step by step. As we will see in the next section, when a loop is closed it is possible to significantly improve the estimation of the map in a single step. In this case, if needed, the absolute map has to be re-computed starting from the seeding features.

In the estimation of the relative map, the inverse of the covariance matrix (the information matrix) is block diagonal and therefore its computational complexity is independent of the number of features. This reflects the fact that the estimation process is local, which is a consequence of the shift and rotation invariance of the estimated state. From a mathematical point of view, the block diagonal structure can be easily deduced starting from the basic equations of the

information filter [20] and by noting that the state is not subjected to any dynamics and the observation consists of a part of the state.

In order to avoid redundancy, only independent invariants are selected from a single observation. For instance, in the case of point features, if a single observation consists of N_{obs} features, the dimension of the observation vector is $2N_{obs} - 3$ (i.e. containing all the independent distances among the observed point features). Obviously, this does not imply that the elements in I are independent since the independency only regards the invariants in a single observation. In particular, when the robot closes a loop, the invariants among the last and the initial observed features are dependent on the ones stored in I before the loop closure. The methods introduced in the next section are able to exploit the information contained in all the dependencies among the invariants in I .

III. HOW TO EXPLOIT THE INFORMATION CONTAINED IN THE RELATIVE MAP DEPENDENCIES

As previously mentioned, the state I in general contains elements which are dependent. In particular, when a loop is closed, the robot observes again the same features observed at the beginning of the loop. The invariants introduced in the state I at this step depend on almost all the other invariants previously introduced in I . Exploiting this dependency is a powerful method to correct the entire relative map. The problem is that, a part a few special cases (for instance if the robot orientation is a priori known), it is impossible to derive these analytical dependencies. The idea is to proceed numerically.

The methods we describe here will apply not necessarily for the dependencies arising at the loop closure. The procedures are very general. After their application, the invariants will be correlated. However, this is not a problem for the computational requirement (e.g. the map dependency constraint can be imposed only once at the end). In the next two subsections we introduce the two methods: the first one is based on the *EKF*, the second one on the *LMA*.

A. *EKF* based Approach

This method is very simple. With respect to the method based on the *LMA* it is less expensive. However, when the global map precision is very low before the loop closure, the second method performs better.

First of all, we select from I a state I_i satisfying the following properties:

- Its elements are independent;
- It allows us to compute the absolute map starting from the absolute location of the seeding features.

The second property allows us to write

$$X = A(I_i) \quad (1)$$

where X is a vector containing the absolute map and A is a function that can be easily implemented.

We stuck all the elements of I , which do not belong to I_i , in to a separate state, I_o . Let us indicate with P_i and P_o the covariance matrices of I_i and I_o , respectively. The state I_o depends on I_i , for construction. Therefore, it is possible to predict I_o by knowing I_i . Let us indicate this prediction with I_o^{pred} . The computation of I_o^{pred} is carried out by computing first of all the absolute map through (1), starting from I_i and the knowledge of the seeding features. Then, each element of I_o^{pred} is computed by selecting the two features defining it. By indicating this functional dependency with B , we can write:

$$I_o^{pred} = B(I_i) \quad (2)$$

We exploit the information contained in I_o to improve the estimation of I_i by applying an *EKF*. In particular, the state estimated by this *EKF* is I_i and the observation is I_o .

The equations to update I_i and its covariance matrix are [1]

$$I_i^{new} = I_i + P_i H^T (H P_i H^T + P_o)^{-1} (I_o - I_o^{pred}) \quad (3)$$

$$P_i^{new} = P_i - P_i H^T (H P_i H^T + P_o)^{-1} H P_i \quad (4)$$

where H is the Jacobian of the function B introduced in (2) with respect to I_i .

The computation of all the elements of I_o^{pred} requires to derive the absolute map only once, and therefore the complexity is $O(N)$.

Regarding the Jacobian we proceed numerically. Let us consider the kl element of this Jacobian. By definition we have:

$$H_{kl} = \lim_{\delta \rightarrow 0} \frac{B(I_i^{l+})_k - B(I_i^{l-})_k}{2\delta} \quad (5)$$

where I_i^{l+} and I_i^{l-} are the vectors obtained from I_i by increasing and decreasing its l^{th} component by δ , respectively.

The numerical computation of H_{kl} can be performed by implementing several robust algorithms (see for instance [18]). The simplest approach uses the definition of the derivative in (5) for some small numerical value of δ ($\delta \ll (I_i)_l$). In order to do this, it is required to compute the equation (2) for $2N_d$ values of I_i , where N_d is the dimension of I_i ($N_d = 2N - 3$ in the case of point features). The complexity of this computation is $O(N^2)$.

This strategy integrates the information coming from the loop closure much better than in the *EKF-SLAM*. Indeed, the *EKF-SLAM* requires to maintain all the correlations among the features and the robot and their computation is affected by the linearization introduced at each step and by an unperfect knowledge about the sensor error model. In contrast, this approach exploits the dependencies among the invariants which reflect the structure (geometry) of the environment (equation (2)).

The other advantage of this strategy is that this filter can be applied only once at the end without any loss of information.

In contrast, the *EKF-SLAM* requires a very expensive computation since the complexity of each step is $O(N^2)$.

B. LMA based Approach

This approach directly estimates the absolute map (i.e. the vector X introduced in (1)) starting from the knowledge of the seeding feature locations and the vector I estimated from the relative map filter and which contains dependent elements. The first step of this method is the initialization of the absolute map (X_{in}). This is obtained through (1).

For a given absolute map X it is possible to compute the observed invariants (I_c). We have

$$I_c = \phi(X) \quad (6)$$

where ϕ is a vector function, mapping the absolute feature configurations into the adopted invariants. For instance, in the case of point features, the components of the function ϕ are the distances among the features which have been observed at least once simultaneously during the robot motion. On the other hand, the relative map filter provides the state I containing the same invariants computed in (6). Starting from the difference state $I_d \equiv I - I_c$ we introduce the cost function

$$c(X) = \| I_d \| \quad (7)$$

The symbol $\| \cdot \|$ is adopted to indicate a suitable norm function. We adopt the following norm:

$$\begin{aligned} \| I - \phi(X) \| &= \\ &= \sum_{i=1}^N \sum_{j=1}^N [I_i - \phi_i(X)] P_{ij}^{-1} [I_j - \phi_j(X)] \end{aligned} \quad (8)$$

where P is the estimated covariance matrix of I .

Now the problem can be formulated as an optimization problem. Starting from the initial absolute map estimation X_{in} , the objective is to minimize the cost function in (7), i.e.:

$$X_{best} = \arg \min [c(X)] \quad (9)$$

To this end, we adopt the Levenberg-Marquardt algorithm [9].

This approach is more expensive than the one based on the *EKF*. On the other hand, it is more robust with respect to the non-linearities and therefore it performs better when the map precision is very low before the loop closure.

IV. RESULTS

In order to carefully evaluate the proposed methods, we compared their performance with the one obtained by implementing the *EKF-SLAM*. Simulations are more suitable for this comparison since the ground truth is known and the comparison can be easily restricted to the estimation process, which is considered in this paper. For data generation and the *EKF-SLAM* we adopted the code provided in [21].

In fig 1 the environment adopted in our simulations is displayed. The adopted unit is meter for both axes. The

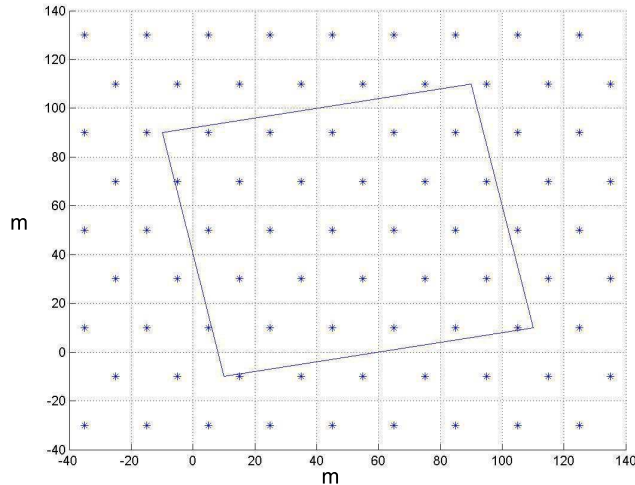


Fig. 1. The considered environment with the actual landmark locations

stars represent the actual landmark locations. The data associations were given. The robot accomplished two loops in counter clockwise direction.

The simulated exteroceptive sensor provides the bearings and the ranges of the landmarks from the robot whose distance does not exceed $30m$. Furthermore, only the landmarks in front of the robot can be seen (i.e. the sensor angle of view is $180deg$). Both the bearings and the distances were generated as Gaussian quantities with variances equal to σ_B^2 and $(0.5m)^2$, respectively. We performed many simulations by varying the value of σ_B in the range $[1, 10]deg$. The frequency was $5Hz$. The robot speed was set to $3\frac{m}{s}$.

Regarding the odometry, which is only required in the case of *EKF-SLAM*, we considered the differential drive. In particular, we modified the code in [21] by introducing an odometry sensor satisfying the model introduced in [2]. According to this model, the translation of the right/left wheel as estimated by the odometry sensors was generated as a Gaussian random quantity satisfying the following relation:

$$\delta\rho^{R/L} = \overline{\delta\rho}^{R/L} + \nu^{R/L} \quad \overline{\delta\rho}^{R/L} = \delta\rho^{aR/L}\delta_{R/L} \quad (10)$$

$$\nu^{R/L} \sim N(0, K|\delta\rho^{aR/L}|)$$

In other words, both $\delta\rho^R$ and $\delta\rho^L$ were assumed Gaussian random variables, whose mean values were given by the actual values (respectively, $\delta\rho^{aR}$ and $\delta\rho^{aL}$) corrected for the systematic errors (which were assumed to increase linearly with the distance travelled by each wheel), and whose variances also increased linearly with the travelled distance. In our simulation we set $K = 0.0001m$, which corresponds to an indoor environment [12] (i.e. where the odometry is very accurate). Furthermore, we assumed an odometry sensor perfectly calibrated (i.e. $\delta_R = \delta_L = 1$ and perfectly known distance between the wheels).

Figs 2-5 show the results obtained for a given simulation with $\sigma_B = 3deg$.

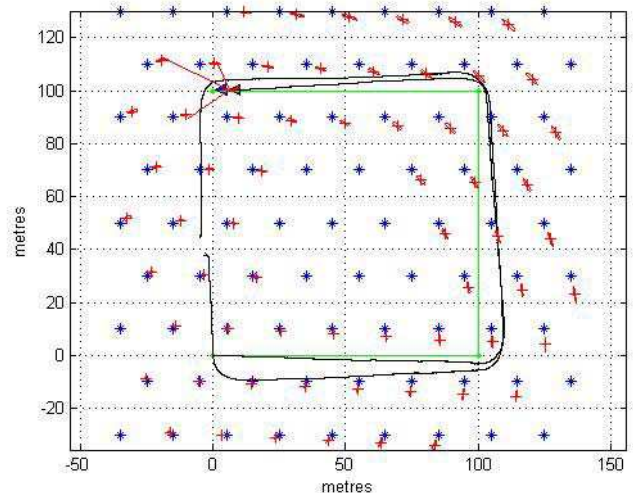


Fig. 2. The estimated landmark locations obtained by implementing *EKF-SLAM* with an odometry sensor perfectly calibrated.

In fig 2 we display the results obtained by implementing the *EKF-SLAM*. The mean error on the estimated landmark locations, defined as the distance between the estimated landmark positions and true landmark positions averaged on all the landmarks, is $E_m = 4.42m$. The total computation time needed for the estimation process is $T_c = 61.7s$. We note that the results get significantly worse as soon as a systematic component is introduced. In particular, we obtained $E_m = 14.6m$ by setting $\delta_L = 1.001$, which still corresponds to an odometry sensor very well calibrated [12].

Fig. 3 shows the results obtained through the relative map filter before applying the loop closure constraint. In this case $E_m = 2.56m$. The computation time needed for the estimation process is much smaller in this case: $T_c = 0.042s$. The results after applying the first proposed method based on the *EKF* are displayed in fig. 4. In this case $E_m = 0.83m$. The time needed for the computation is $T_c = 1.21s$. Finally, in fig. 6 the results obtained from the second method based on the *LMA* are shown. The precision is even better than in the previous case, being $E_m = 0.38m$. The computational time increases ($T_c = 4.76s$).

We remark that in our simulations we adopted a very accurate odometry, which is definitely unrealistic especially for an outdoor environment. This significantly improved the performance of the *EKF-SLAM*.

We performed many simulations and we found that the value of E_m can change a little bit in contrast with the value of T_c which is almost constant. This especially holds for the *EKF-SLAM* and the relative map approach (before imposing the loop constraint). For this reason, we performed for each value of σ_B in the range $[1, 10]deg$, 20 independent simulations. In fig 6 we plot the values of E_m averaged on these simulations. We display the results for $\sigma_B \leq 5deg$. For larger values, the first method sometimes diverges, since the error before the loop closure is too big. In contrast, the

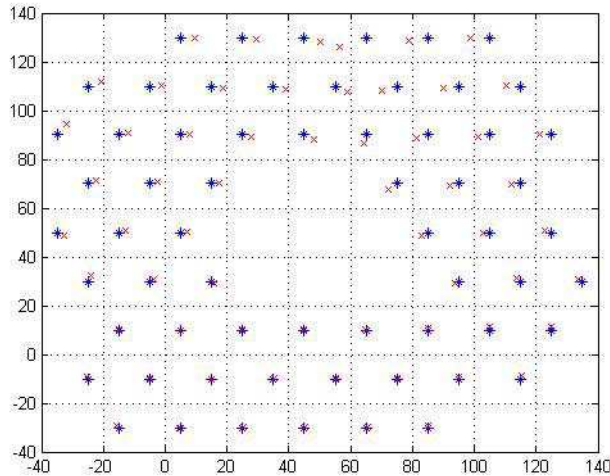


Fig. 3. The estimated landmark locations (red \times) obtained by implementing the relative map filter before exploiting the information at the loop closure

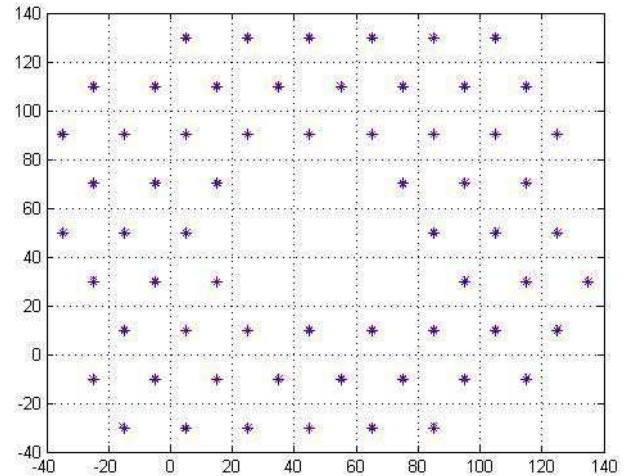


Fig. 5. The estimated landmark locations (red \times) obtained by implementing the relative map filter after exploiting the information at the loop closure by means of the second method based on the *LMA*.

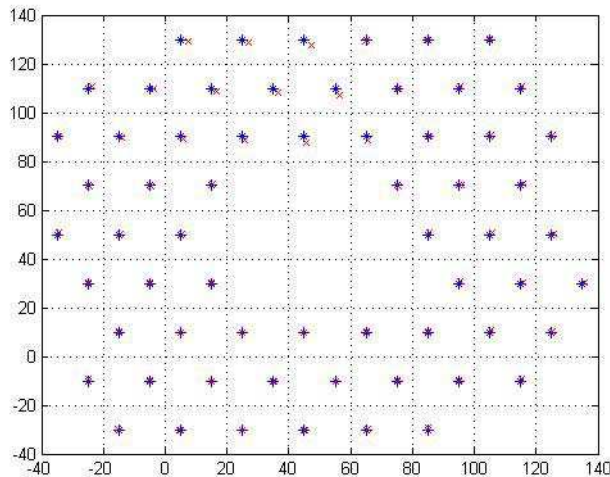


Fig. 4. The estimated landmark locations (red \times) obtained by implementing the relative map filter after exploiting the information at the loop closure by means of the first method based on the *EKF*.

method based on the *LMA*, performs very well. In particular, the error still grows linearly as in the range $\sigma_B \in [0, 5]deg$.

V. CONCLUSIONS

We introduced and discussed two methods able to exploit the information at the loop closure. The methods apply to a relative map approach to *SLAM* and are able to exploit the information contained in all the dependencies among the elements belonging to the relative state, which is shift and rotation invariant.

The methods have three fundamental advantages:

- 1) they do not require to compute any correlation among the features which are not observed simultaneously to apply the loop closure constraint;

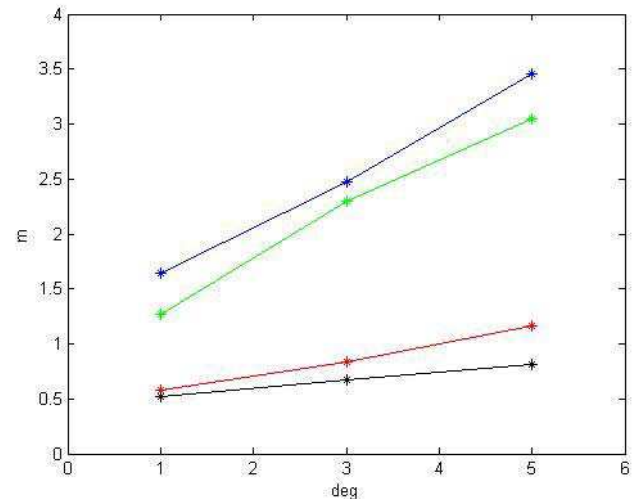


Fig. 6. Values of E_m averaged on 20 simulations vs the bearing standard deviation σ_B . Green line refers to the *EKF-SLAM*, blue line to the relative map filter without the loop constraint, red line to the first method based on the *EKF* and black line to the second method based on the *LMA*.

- 2) the loop closure constraint can be applied only once at the end (even after more than one loop) in a single step;
- 3) the linearization does not affect the estimation process (this especially holds for the second method which is based on the Levenberg-Marquardt algorithm).

The first two advantages make the complexity of the methods during the robot exploration independent of the number of features. The third one makes the final estimation precision significantly better than the one achievable by implementing the conventional *EKF* based *SLAM*. Indeed,

EKF – SLAM needs correlations among the features in order to exploit the information at loop closure. The computation not only is very expensive but also approximated since affected by the linearization and by an approximated knowledge of the sensor error model. In contrast, the proposed methods exploit geometrical constraints directly and simply derived from the structure of the environment.

The methods can be used with any kind of features.

Simulations show that both methods outperform significantly the conventional *EKF* based *SLAM* both in computational cost and accuracy.

We are currently implementing both methods in real experiments.

ACKNOWLEDGMENT

This work has been supported by the European project *BACS* (Bayesian Approach to Cognitive Systems) and by the Swiss National Science Foundation.

The authors wish also to thank Tim Bailey for making available on line the *EKF – SLAM* code.

REFERENCES

- [1] Bar-Shalom, Li X.R. and Kirubarajan T., Estimation with Applications To Tracking and Navigation. John Wiley and Sons, Inc., 2001
- [2] Borenstein J., Feng L., "Measurement and correction of systematic odometry errors in mobile robots," *IEEE Transactions on Robotics and Automation*, vol. 12, pp. 869–880, 1996.
- [3] Castellanos, J. A., Neira, J. and Tardós, J.D. (2004). Limits to the consistency of the EKF-based SLAM. in Intelligent Autonomous Vehicles (IAV 2004), Lisbon, Portugal, July 2004.
- [4] Crowley, J.L., (1989). World Modeling and Position Estimation for a Mobile Robot Using Ultrasonic Ranging. IEEE International Conference on Robotics and Automation (ICRA), Scottsdale, AZ.
- [5] M.Csorba, J.K.Uhlmann and H.F.Durrant-Whyte "A subOptimal Algorithm For Automatic Map Building," *American Control Conference*, p. 537 -541, Albuquerque, New Mexico, USA 1997.
- [6] M.C.Deans and M.Hebert, "Invariant Filtering for Simultaneous Localization and Mapping," *International Conference on Robotics and Automation*, vol. 2, pp. 1042 -1047, San Francisco, CA, USA, April 2000.
- [7] Dissanayake, Newman, Clark, Durrant-Whyte and Csorba, 2001, A Solution to the Simultaneous Localization and Map Building (SLAM) problem, *IEEE Trans. On Rob. And Aut.* Vol 17, No.3, June 2001
- [8] Alfredo Germani, Costanzo Manes and Pasquale Palumbo, "Polynomial Extended Kalman Filter", *IEEE Trans. on Automatic Control*, Vol. 50, No. 12, pp. 2059-2064, 2005;
- [9] Gill, P. E. and Murray, W. "Algorithms for the solution of the nonlinear least-squares problem", *SIAM J. Numer. Anal.* 15 [5] 977-992, 1978
- [10] S.Julier and J.K.Uhlmann, 2001, A Counter Example to the theory of Simultaneous Localization and Map Building. International Conference on Rob. and Aut. (ICRA01), Seoul, Korea, 2001
- [11] J.J. Leonard, H.F. Durrant-Whyte, "Directed Sonar Sensing for Mobile Robot Navigation," *Kluwer Academic Publishers*, Dordrecht, 1992.
- [12] Martinelli A, Tomatis N, Tapus A. and Siegwart R., "Simultaneous Localization and Odometry Calibration" *International Conference on Intelligent Robot and Systems (IROS03)* Las Vegas, USA
- [13] Martinelli A, Tomatis N and Siegwart R., "Open Challenges in SLAM: An Optimal Solution Based on Shift and Rotation Invariants" *International Conference on Rob. and Aut. (ICRA04)* New Orleans, USA, 2004.
- [14] A. Martinelli, V. Nguyen, N. and R. Siegwart R., "A Relative Map approach to SLAM Based on Shift and Rotation Invariants" to appear on the *Robotics and Autonomous Systems Journal*.
- [15] P.M.Newman, "On the Structure and Solution of the Simultaneous Localization and Mapping Problem," *PhD thesis*, Australian Centre for Field Robotics, University of Sydney, 1999.
- [16] P.M.Newman and H.F.Durrant-Whyte, "An efficient solution to the slam problem using geometric projection," *Sensor Fusion and Decentralized Control in Robotics Systems*, Boston, USA, 2001.
- [17] Nguyen, V.T., Martinelli A and Siegwart R., "Handling the Inconsistency of Relative Map Filter" *International Conference on Rob. and Aut. (ICRA05)* Barcelone, Spain, 2005.
- [18] Press, W. H.; Flannery, B. P.; Teukolsky, S. A.; and Vetterling, W. T. "Numerical Derivatives." 5.7 in *Numerical Recipes in FORTRAN: The Art of Scientific Computing*, 2nd ed. Cambridge, England: Cambridge University Press, pp. 180-184, 1992.
- [19] Smith, Self, et al. (1988) "Estimating uncertain spatial relationships in robotics" *Uncertainty in Artificial Intelligence 2* Elsevier Science Pub: 435-461.
- [20] S. Thrun, W. Burgard, and D. Fox "Probabilistic Robotics". MIT-Press, 2005.
- [21] <http://www.acfr.usyd.edu.au/homepages/academic/tbailey/software/index.html>, *EKF – SLAM* version 1