Optimal local map size for EKF-based SLAM

Lina M. Paz, José Neira
Computer Science Dept. of the University of Zaragoza,
María de Luna 1, 50018 Zaragoza (Spain).
E-mail: {linapaz, jneira}@unizar.es

Abstract— **In this paper we show how to optimize the computational cost and maximize consistency in EKF-based SLAM for large environments. We combine Local Mapping with Map Joining in a way that the total cost of computing the final map is minimized compared to full global EKF-SLAM. This solution is not now only shown to be (1) *computationally optimal*, but in addition, it is empirically shown that (2) it also produces the *most consistent environment map*. For a given environment size and sensor range, we can determine the optimal size of the local maps required to minimize the total computational cost and maximize map consistency. The motivation of this work is described in a map building experiment in our lab, and the statistical significance of the proposed method is validated using Monte Carlo simulations.**

Index Terms— **EKF SLAM, Local Mapping, Map Joining, Computational Cost, Map Consistency**

I. Introduction

The problem of Simultaneous Localization and Mapping (SLAM) consists in placing a vehicle in an unknown environment with no prior information on its location, and have it concurrently determine both its location and the environment structure. SLAM is considered essentially solved for small environments. Both Extended Kalman Filter based methods (EKF-SLAM) [1], [2], [3] and sampling based methods [4], [5] can robustly and efficiently deal with environments of limited size.

In recent years, interest has focused on reducing the computational cost of performing SLAM in large environments. EKF-SLAM techniques have a *first* important limitation: the computational cost of updating the map at each step is of order $O(n^2)$, where n is the number of features in the map. In sampling based methods, the number of required samples grows exponentially with the uncertainty of the vehicle location. In the case of the EKF, the major computational costs involved is updating the $n \times n$ covariance matrix. The Sparse Extended Information Filter (SEIF) [3] addresses the complexity issue of the EKF by observing that the information matrix, the inverse of the covariance matrix computed by the EKF, is approximately sparse and thus $O(n)$ updates can be achieved by working on a sparsified approximation of the information matrix. However, it has been pointed out [6] that the EKF-based estimation has a *second* important limitation: it quickly becomes inconsistent in large environments due to nonlinearities inherent to most SLAM problems. Thus SEIFs estimations will suffer from the same inconsistency problems of EKF.

In this paper we study an alternative technique that both reduces the computational cost and improves consistency: Local Map Sequencing [14]. Instead of building one global map, this technique builds a set of independent local maps of limited size. Local maps are then joined together into a global map that should be equivalent to the map obtained by the standard EKF-SLAM approach, except for linearization errors. How to decide the size of each local map is an important issue that remains unanswered in [14]. Related decompositions have been proposed in [7], [8], [9], [10]. There are also a number of methods that use local maps for SLAM [11], [12], [13] with alternative joining steps. None of these methods answer this question. We believe that these methods can also benefit in terms of computation cost as well as map consistency from the analysis proposed here.

To compare local map sequencing with full EKF-SLAM, we performed a map building experiment, using a robotized wheelchair equipped with a SICK laser scanner. The vehicle was hand-driven along a loop of about 250 m in a populated indoor/outdoor environment in the Ada Byron building of our campus. The laser scans were segmented to obtain lines using the RANSAC technique. The global map obtained using the classical EKF-SLAM algorithm is shown in figure 1, top. At this point, the vehicle was very close to the initial starting position, closing the loop. The figure shows that the vehicle estimated location has some 10m error and the corresponding 95% uncertainty ellipses are ridiculously small, giving an inconsistent estimation. Due to these small uncertainty, the data association algorithm was unable to properly detect the loop closure. This corroborates the results obtained with simulations in [6]: in large environments the map obtained by EKF-SLAM quickly becomes inconsistent, due to linearization errors.

What happens if we divide the full map in N smaller local maps and then join them together? In order to try to answer this question, the same data set was processed to obtain 25 independent local maps at fixed trajectory intervals of about 10 m. The local maps were joined and fused, obtaining the global map shown in figure 1, bottom. In this case the loop was correctly detected by data association and the map obtained seems to be consistent. Furthermore, the computational time was about 50 times smaller than the standard EKF approach. Thus, there is a strong reduction in computational cost and increase in consistency in the resulting map in local map sequencing.

This experiment raises the following questions: given a certain environment size and sensor characteristics (1) is there a *computationally optimal* local map size, such that the total

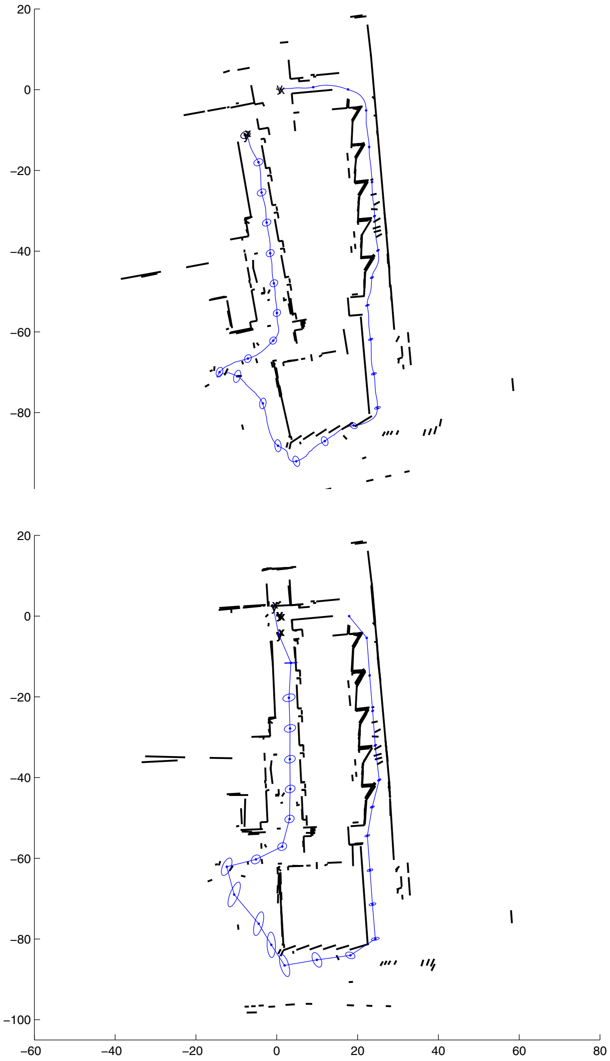

Fig. 1. Global maps obtained using the standard EKF-SLAM algorithm (top) and local map joining for $N = 25$ local maps(bottom).

computational cost is minimal? (2) is there a *consistency optimal* local map size, such that the consistency of the final map is maximal? (3) do these optimals coincide?

In this paper we try to answer these questions. In section II we describe the local map sequencing problem for a given environment size and feature density, sensor characteristics and size of local map. In section III we detail its related computational cost and derive the optimal solution. We validate this solution using a simulated experiment. In section IV, we study the obtained solution with respect to map consistency and error. Monte Carlo simulations are used to show that the solution of minimal computational cost is also the one that provides the map of maximal consistency. Finally, in section V we draw the main conclusions of this work and outline future directions of work.

II. Problem Definition

In order to analyze the computational cost involved in performing SLAM using local map sequencing, we consider a simulated exploration experiment with the following assumptions: the environment to be mapped consists in P equally spaced features $\{\mathbf{f}_1 \cdots \mathbf{f}_P\}$. We are provided with a sensor whose range allows to obtain measurements of m of the environment features at a time in this environment. We carry out a straight forward exploration trajectory (with no loop closing) in which we stop to obtain measurements at equally spaced intervals. This allows to reobserve r features seen in the previous step, i.e., at each step there will be $n = m - r$ measurements corresponding to features previously unseen. At step 1 measurements in the first measurement set $\mathbf{z}_1$ will correspond to features $1 \ldots m$, measurements $\mathbf{z}_2$ at the second step will correspond to features $n+1 \ldots n+m$, and thus at step i, the last measurement of $\mathbf{z}_i$ will correspond to feature $\mathbf{f}_{f(i)}$, where:

$$\begin{aligned} f(i) &= (i-1)n + m \\ &= (i-1)(m-r) + m \end{aligned} \qquad (1)$$

If we decide to make a local maps of size p features, we will reach feature $\mathbf{f}_p$ at step i, where:

$$p = (i-1)(m-r) + m$$

and thus:

$$i = \frac{p-m}{m-r} + 1 \qquad (2)$$

This means that in order to have p features in local map $\mathbf{m}_1$, it should be built using steps $1, \ldots, \frac{p-m}{m-r} + 1$, the second local map $\mathbf{m}_2$ should be built using measurements from steps $\frac{p-m}{m-r} + 2, \ldots, 2\frac{p-m}{m-r} + 2$, and in the same way, map $\mathbf{m}_k$ will be built using measurements at step $s(k)$, $\mathbf{z}_{s(k)}$, where:

$$s(k) = k\left(\frac{p-m}{m-r}\right) + k \qquad (3)$$

Thus, to determine which feature we will reach in local map $\mathbf{m}_k$, we substitute i for $k\left(\frac{p-m}{m-r}\right) + k$ in eq. (1):

$$\begin{aligned} f(s(k)) &= \left(k\left(\frac{p-m}{m-r}\right) + k - 1\right)(m-r) + m \\ &= kp - (k-1)r \end{aligned} \qquad (4)$$

This means that if we want to reach the last environment feature $\mathbf{f}_P$ in the last step of map $\mathbf{m}_N$ and thus conclude mapping, according to eq. (4) we will have:

$$P = Np - (N-1)r$$

and thus the number of local maps N will be:

$$N = \frac{P-r}{p-r} \qquad (5)$$

Consider as an example an experiment in an environment with $P = 270$ total point features, 135 on each side of the straight path that the vehicle traverses. Features are $2m$ apart.

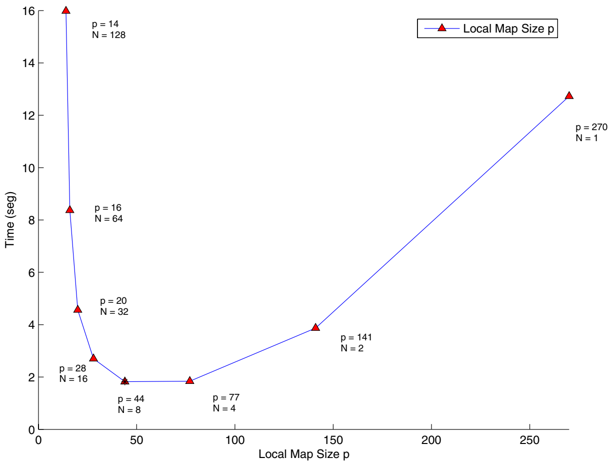

Fig. 2. Execution time in seconds in Matlab of Local Maps Sequencing for local maps from $p = 14$ to $p = 270$. Results for N=1,2,4,8,16,32,64 and 128 local maps are shown (correspondingly, p=270,141,77,44,28,20,16,14).

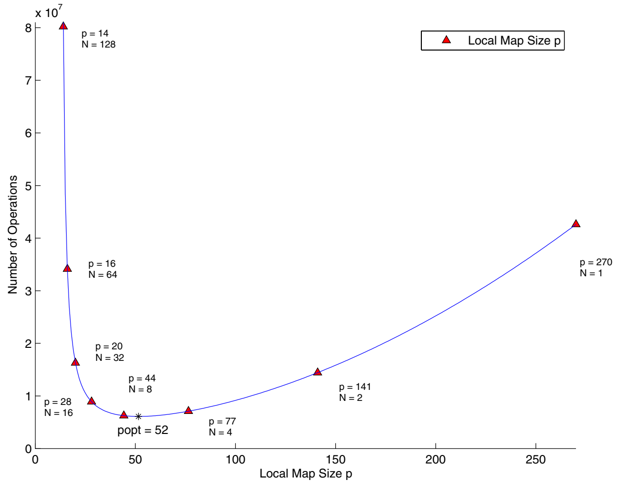

Fig. 3. Computational cost versus local map size p in number of operations according to C_{TOTAL}; results for N=1,2,4,8,16,32,64 and 128 local maps are shown (correspondingly, p=270,141,77,44,28,20).

The range and bearing sensor has a range such that $m = 14$ features are perceived from a certain location. The robot moves $2m$ at each step. The vehicle thus leaves $n = 2$ features behind at each step and thus reobserves $r = 12$. If we decide to have local maps of size $p = 270$ then obviously $N = 1$; if we choose the smallest local map size of $p = m = 14$, then we should build $N = 128$ local maps.

We programmed the Local Map Sequencing algorithm of [14] in Matlab and executed the mapping process for different sizes of local maps: from the smallest value of $p = 14$ requiring $N = 128$ local maps, to building one local map of size $p = 270$ covering the whole environment. This last extreme is equivalent to full EKF-SLAM.

Figure 2 shows the resulting execution times for the different values of p from 14 to 270. We can see that there is an *optimal* value for the local map size p (in this case close to $p = 44$) that minimizes the total cost of SLAM. Using this optimal value we can greatly reduce the cost of computation compared to full EKF-SLAM (dividing by 7 in this small example). In the next section we will derive the expression for the computational cost of Local Map Sequencing that allows to determine the optimal value of p for a given mapping problem.

III. Computational cost of EKF-SLAM

A. Local maps with Standard EKF-SLAM

Consider building a map $\mathbf{m} = (\hat{\mathbf{x}}, \mathbf{P})$ containing p features for the experiment proposed above. At the beginning of step k, when the vehicle moves from position $k-1$ to k, the state vector $\hat{\mathbf{x}}_{k-1}$ and its corresponding covariance matrix $\mathbf{P}_{k-1}$ contain $n_{k-1} = f(k-1)$ features. According to eq. (1):

$$n_{k-1} = (k-2)(m-r) + m$$

During the *prediction step*, the fundamental cost involves updating the correlation between the predicted vehicle location and the n_{k-1} map features. Hence, we consider the cost of the prediction step in number of operations as:

$$C_{Pred_k} = n_{k-1} \qquad (6)$$

During the *estimation step* at k, the sensor obtains a set of measurements of m environment features. We summarize the main matrix operations and the cost necessary to update the estimate according to the number of reobserved features r:

$$\begin{array}{lr} \mathbf{S}_k = \mathbf{H}_k \mathbf{P}_{k|k-1} \mathbf{H}_k^T + \mathbf{R}_k & r n_{k-1}\ ops \\ \mathbf{K}_k = \mathbf{P}_{k|k-1} \mathbf{H}_k^T (\mathbf{S}_k)^{-1} & r^2 n_{k-1}\ ops \\ \hat{\mathbf{x}}_k = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k(\mathbf{z}_k - \mathbf{h}_k(\hat{\mathbf{x}}_{k|k-1})) & r n_{k-1}\ ops \\ \mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k)\mathbf{P}_{k|k-1} & r n_{k-1}^2\ ops \end{array}$$

Thus, the total cost of the update step is:

$$C_{Up_k} = r n_{k-1} + r^2 n_{k-1} + r n_{k-1} + r n_{k-1}^2$$

During the *addition step*, n new features are directly added to the current stochastic state vector $\hat{\mathbf{x}}_k$ by using the relative transformation between the robot R_k and the observed features, which requires n operations. Similarly, the covariance matrix $\mathbf{P}_k$ will require n^2 operations to add $\mathbf{R}_k$, and $n_{k-1}n$ to update correlations among the robot and the new features. The main cost of adding new features is summarized as follows:

$$C_{Ad_k} = n_{k-1} n + n^2$$

Carrying out EKF to build a map of p features requires that we process steps 1 to $\frac{p-m}{m-r} + 1$. The total cost associated will be:

$$\begin{aligned} C_{1map} &= C_{Pre_k} + C_{Up_k} + C_{Ad_k} \\ &= \sum_{k=1}^{\frac{p-m}{m-r}+1} (n_{k-1} + r n_{k-1} + r^2 n_{k-1} \\ &+ r n_{k-1} + r n_{k-1}^2 + s n_{k-1} + s^2) \end{aligned}$$

Solving the arithmetic progression for all steps we obtain the total cost of building a local map of p features as (see the values from k_1 to k_4 in the appendix):

$$C_{1map} \;=\; k_1 p^3 + k_2 p^2 + k_3 p + k_4 \tag{7}$$

B. EKF-SLAM with Local Map Sequencing

The standard EKF-SLAM technique presented above is used in Local Map Sequencing to divide the full map in N local maps $\mathbf{m}^1 \ldots \mathbf{m}^N$ containing p features each. In order to map the environment up to the last feature $\mathbf{f}_P$, $N = \frac{P-r}{p-r}$ local maps should be built. The cost of building those N maps is given by (see appendix for the values of k_5 to k_8):

$$\begin{aligned} C_{Nmap} &= \left(\frac{P-r}{p-r}\right) C_{1map} \\ C_{Nmap} &= k_5 p^2 + k_6 p + k_7 + k_8 \frac{1}{(p-r)} \end{aligned} \tag{8}$$

The N local maps are joined into global map $\mathcal{M} = \left(\hat{\mathcal{X}},\, \mathcal{P}\right)$ in $j = 1 \ldots N-1$ join and optimize steps, similar to those of EKF (please refer to [14] for the complete formulation). At global step j, we join global map $\mathcal{M}_j = \left(\hat{\mathcal{X}}_j,\, \mathcal{P}_j\right)$ with local map $\mathbf{m}^{j+1} = \left(\hat{\mathbf{x}}^{j+1},\, \mathbf{P}^{j+1}\right)$. According to eq. (4), the size of global map $\mathcal{M}_j$ is:

$$\begin{aligned} n_j &= f(s(j)) \\ &= jp - (j-1)r \end{aligned}$$

The *joining step* consists in obtaining the state vector $\hat{\mathcal{X}}_{j+1/j}$ and covariance matrix $\mathcal{P}_{j+1/j}$. In the joined state vector $\hat{\mathcal{X}}_{j+1/j}$, all p features in local map $\mathbf{m}_j$ must be referenced to the global frame; this requires p operations. The covariance matrix $\mathcal{P}_{j+1/j}$ update requires computing the correlations between the $jp + (j-1)r$ features in the global map and the p features in the local map, for a cost of:

$$C_{MJ_j} = n_j\, p \tag{9}$$

We apply the specialized EKF update equations [14] to obtain a new global map $\mathcal{M}_{j+1} = \left(\hat{\mathcal{X}}_{j+1},\, \mathcal{P}_{j+1}\right)$ as follows:

$$\begin{array}{ll} \mathcal{S}_{j+1} = \mathcal{H}_{j+1}\mathcal{P}_{j+1/j}\mathcal{H}_{j+1}^T & r(n_j+p)\ ops \\ \mathcal{K}_{j+1} = \mathcal{P}_{j+1/j}\mathcal{H}_{j+1}^T(\mathcal{S}_{j+1})^{-1} & r^2(n_j+p)\ ops \\ \hat{\mathcal{X}}_{j+1} = \hat{\mathcal{X}}_{j+1/j} - \mathcal{K}_{j+1}\mathbf{h_{j+1}}(\hat{\mathcal{X}}_{\mathbf{j+1/j}}) & r(n_j+p)\ ops \\ \mathcal{P}_{j+1} = (\mathbf{I} - \mathcal{K}_{\mathbf{j+1}}\mathbf{H}_{\mathcal{H}})\mathcal{P}_{\mathbf{j+1/j}} & r(n_j+p)^2\ ops \end{array}$$

Thus, the total cost of an update is:

$$\begin{aligned} C_{MJU_j} &= r(n_j+p) + r^2(n_j+p) + \\ & \quad r(n_j+p) + r(n_j+p)^2 \end{aligned}$$

Finally, the total cost of Local Map Sequencing is

$$\begin{aligned} C_{TOTAL} &= C_{Nmap} + \sum_{j=2}^{N} C_{MJU_j} \\ &= \frac{k_9 p^4}{(p-r)^2} + \frac{k_1 0p^3}{(p-r)^2} + \\ & \quad + \frac{k_{11}p^2}{(p-r)^2} + \frac{k_{12}\,p}{(p-r)^2} + \\ & \quad + \frac{k_{13}}{(p-r)^2} + \frac{k_{14}\,p^3}{(p-r)} + \frac{k_{15}\,p^2}{(p-r)} + \\ & \quad + \frac{k_{16}\,p}{(p-r)} + \frac{(k_{17}+k_8)}{(p-r)} + (k_{18}+k_5)p^2 + \\ & \quad + (k_{19}+k_6)p + (k_{20}+k_7) \end{aligned} \tag{10}$$

See the values from k_9 to k_{20} in the appendix.

C. Optimizing the Total Computational Cost

Consider again the example an experiment in an environment with $P = 270$ total point features, 135 on each side of the straight path that the vehicle traverses. Features are $2m$ apart. The range and bearing sensor has a range such that $m = 14$ features are perceived from a certain location, with a standard deviation error of 5% of the distance in range, and $1deg$ in bearing. The robot moves $2m$ at each step, with standard deviation error of $20cm$ frontal, $10cm$ lateral, and $2deg$ angular. The vehicle thus leaves $n = 2$ features behind at each step and thus reobserves $r = 12$. Using these values, equation 10 becomes:

$$\begin{aligned} C_{TOTAL} &= \frac{-25p^4}{(p-r)^2} + \frac{3369p^3}{(p-r)^2} + \frac{865290p^2}{(p-r)^2} + \\ & \quad + \frac{13521924p}{(p-r)^2} + \frac{-182337480}{(p-r)^2} + \frac{-12p^3}{(p-r)} + \\ & \quad + \frac{948p^2}{(p-r)} + \frac{381816p}{(p-r)} + \frac{63996488}{(p-r)} + \\ & \quad + 528p^2 + 16142p - 247935 \end{aligned}$$

The optimal size of the local maps can be obtained by solving the following equation:

$$\frac{\partial C_{TOTAL}}{\partial p} = 0$$

$$982p^4 - 15638p^3 - 285498p^2 - 97445780p + 9.974e8 = 0$$

Fig. 3 shows the corresponding function C_{TOTAL} for the considered example. The analytical solution for this case can be easily obtained by means of the MATLAB `solve` *function*. According to `solve`, the predicted optimal size is attained when the local map size is $p = 52$ features. $N = 7$ to $N = 8$ local maps should be built and joined. It can be easily seen that C_{TOTAL} resembles very closely the actual execution times of the program in fig. 2, and that the minimum $p = 44$ is very close to the predicted minimum of $p = 52$, coinciding

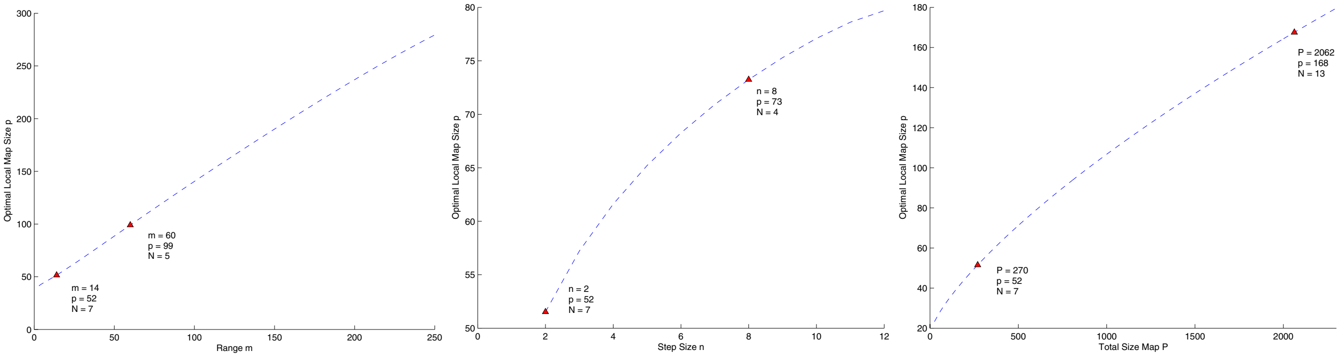

Fig. 4. Optimal size of local maps p: apparent linear growth with the range sensor m (left); sublinear growth with the number of new features per step n (center); and also with the total environment size P (right).

in the optimal number of $N = 8$ local maps. Differences in both plots can be due to memory management and other implementation issues in Matlab.

It is also interesting to study the expected behavior of the optimal computational cost as a function of sensor characteristics and total map size. For this purpose, we computed the optimal size p as a function of the sensor range m (fig. 4, left), of the number of new features per step n (center), and of the size of the environment P (right). The optimal local map size show an apparent linear growth with m, and sublinear growth with n and P.

IV. Vehicle and Map Consistency

In EKF-SLAM, the computational cost depends neither on sensor nor vehicle error. But *map consistency*, how well the computed map covariances $\mathbf{P}$ match the actual estimation errors $\mathbf{x}-\hat{\mathbf{x}}$, greatly depends on sensor and vehicle error. Given that SLAM is a nonlinear problem, consistency checking is of paramount importance. When the ground true solution $\mathbf{x}$ for the state variables is available, a statistical test for filter consistency can be carried out on the Normalized Estimation Error Squared (NEES), defined as:

$$D^2 = (\mathbf{x} - \hat{\mathbf{x}})^T \ \mathbf{P}^{-1} \ (\mathbf{x} - \hat{\mathbf{x}}) \tag{11}$$

Consistency is checked using a chi-squared test:

$$D^2 \leq \chi^2_{r,1-\alpha} \tag{12}$$

where $r = dim(\mathbf{x})$ and α is the desired significance level (usually 0.05). If we define the consistency index of a given estimation $(\hat{\mathbf{x}}, \mathbf{P})$ with respect to its true value $\mathbf{x}$ as:

$$CI = \frac{D^2}{\chi^2_{r,1-\alpha}} \tag{13}$$

when $CI < 1$, the estimation is consistent with ground truth, and when $CI > 1$, the estimation is inconsistent (optimistic) with respect to ground truth.

In order to study the effect to the local map size in the consistency of the resulting map estimate, we carried out Monte Carlo runs on the simulated experiment with the given standard deviation error of 5% of the distance in range, and $1deg$ in bearing for the sensor, and standard deviation error of $20cm$ frontal, $10cm$ lateral, and $2deg$ angular in the vehicle motions. We ran the algorithm 20 times for each possible value of p, from 14 (equal to the sensor range m) to 270 (equal to the environment size P). Figure 5 shows the mean consistency index of the final map for the different local map sizes. Surprisingly, the most consistent resulting map on average coincides with the local map size that is optimal from the computational point of view. It is currently unclear why this happy coincidence occurs. Maybe the large and small linearizations have a least significant effect when the global and local map updates are optimally combined. This will be the subject of future study.

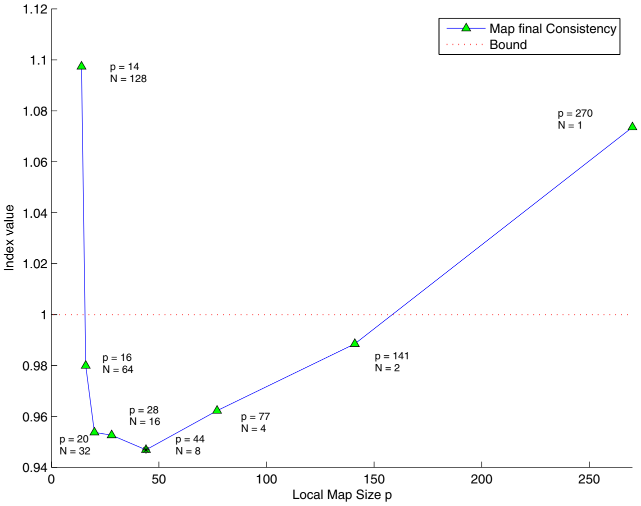

Fig. 5. Map Consistency index with respect to local map size p.

We have also studied the resulting map quality in terms of vehicle and feature error. Fig. 6, top, shows the evolution of the mean absolute angular error of the vehicle. All steps have been executed as the map size increases for both, full EKF-SLAM and Local Map Sequencing algorithms. The figure also shows the theoretical uncertainty (2σ bounds with no measurement or motion noise) of both algorithms. In the case of the full EKF, the computed uncertainty quickly falls both below the

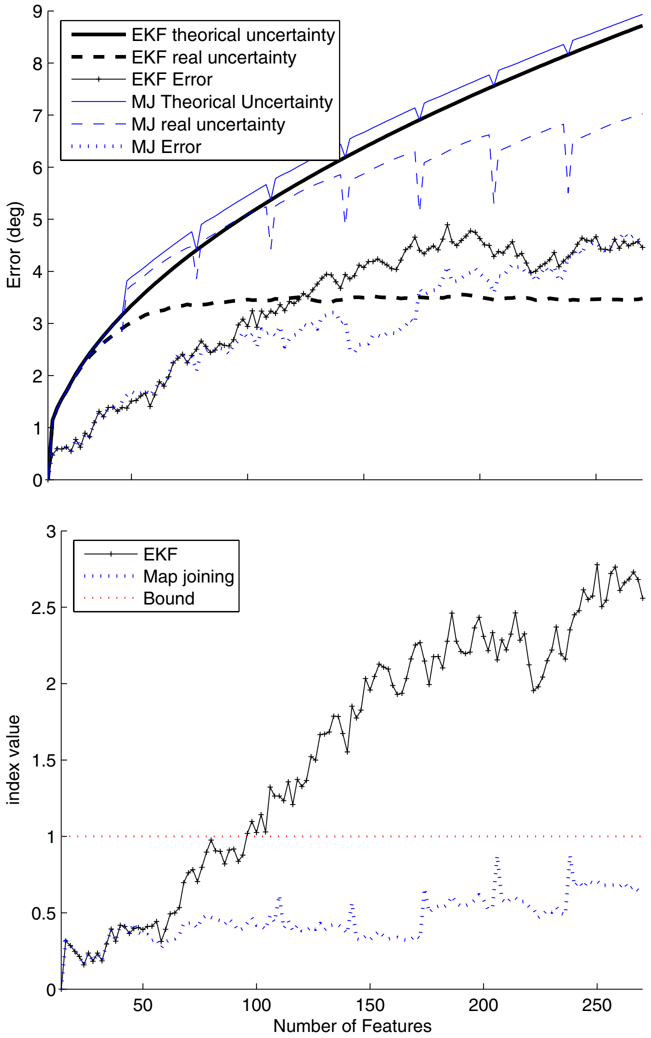

Fig. 6. Vehicle orientation Mean Absolute Errors along the vehicle trajectory for the approaches discussed in the paper: a full EKF and a map joining of 8 local maps. Error and uncertainty bounds (top); Consistency index (bottom).

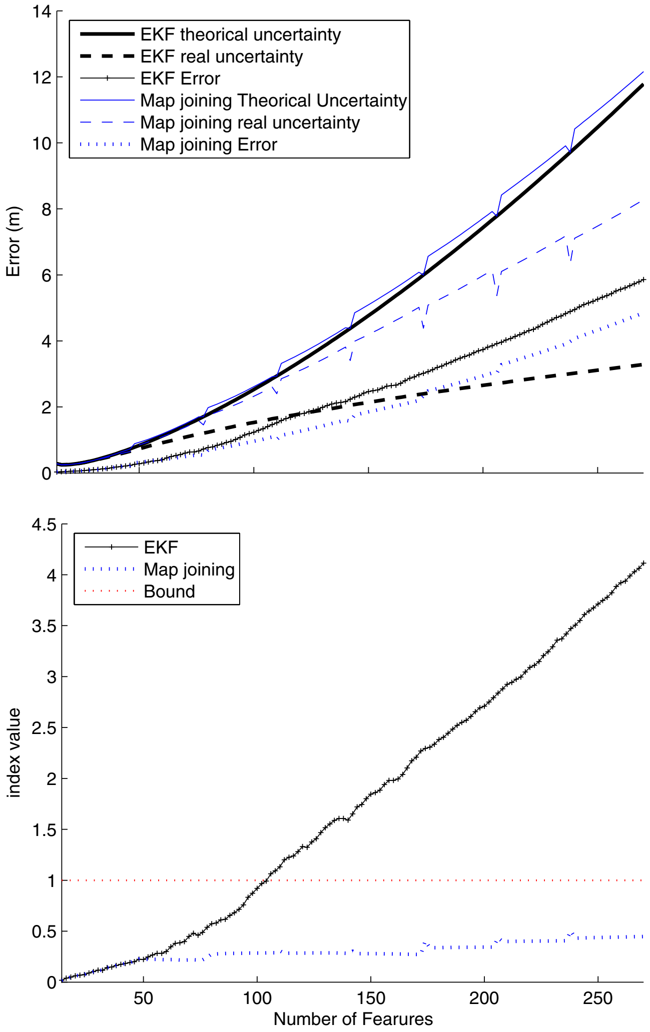

Fig. 7. Position Mean Absolute Errors for all features in the map.

theoretical uncertainty and the actual vehicle error, rendering the estimation useless very quickly. The computed uncertainty (with motion noise) of the Local Map Sequencing also falls below the theoretical uncertainty, but at a much slower rate, and is always over the real angular error for this simulated experiment. Fig. 6, bottom, shows the evolution of the mean consistency index of eq. (13) of the angular error. The estimated vehicle orientation quickly becomes inconsistent for full EKF-SLAM Algorithm compared with Local Map Sequencing Algorithm, whose estimate remains consistent over the full experiment.

Similar results are observed in the case of map feature error. Fig. 7, top, shows the comparative mean error in the y coordinate (orthogonal to the vehicle motion) of all map features for full EKF-SLAM and Local Map Sequencing. The figure also shows the mean 2σ bounds of the theoretical (with no measurement or motion noise) and computed uncertainty (with noise) of both algorithms. It can be seen that the mean feature error of Local Map Sequencing is always smaller than that of full EKF-SLAM. The computed uncertainty of both algorithms falls below the theoretical uncertainty, but at a much slower rate in the case of Local Map Sequencing. Fig. 7, bottom shows the evolution of the mean value of the consistency index for all features as the map grows. Full EKF-SLAM quickly becomes inconsistent, while the estimation obtained by Local Map Sequencing remains consistent for the whole map building process.

V. Conclusions

In this work we have developed computational analysis for EKF-SLAM using local mapping and map joining, that allows to determine the optimal size of local maps to minimize the total computational cost for some given environment size, and sensor and vehicle characteristics. Monte Carlo simulated experiments show that building a global map via local maps and map joining also renders an improvement in linearization errors and thus consistency. In our simulated experiments, the maximum saving achieved with map joining coincides with the maximum of map consistency. This fact can be generalized if the features considered are randomly distributed with a uniform density. At the moment we are carrying out a complementary analysis of consistency to determine why both the computational cost and the consistency are optimized for the same local map size. We feel that other methods based on local maps, besides map joining, can profit from this result to reduce computational cost and overcome nonlinearity problems.

We are currently developing an extension of these methods to work with environments of unknown size. Another issue of interest is to establish how the optimal size of local maps is related with the process noise and the sensor noise. New experiments will be considered using a iterative EKF to analyze these changes. A generalization to different sensor and features types will be studied. These will be the main subjects of our future work.

APPENDIX

Constants for Local mapping

$$
\begin{aligned}
c_1 &= r(m-r)^2 \\
c_2 &= (m-r)\left[(r+1)^2+(m-r)-2r(m-2r)\right] \\
c_3 &= \left[(r+1)^2(2r-m)+r(m-r)+r(2r-m)^2\right] \\
k_1 &= \frac{c_1}{3(m-r)^3} \\
k_2 &= \left[\frac{c_1(m^2+3m-9r)}{6(m-r)}+\frac{c_2}{2}\right]\frac{1}{(m-r)^2} \\
k_3 &= \left[\frac{c_1(5r^2-8rm)}{6(m-r)}+\frac{c_2(m-3r)}{2}\right]\frac{1}{(m-r)^2} \\
k_4 &= \left[\frac{c_1(rm(r-m)+6r^3)}{6(m-r)}+\frac{c_2(2r^2-rm)}{2}\right]\frac{1}{(m-r)^2}
\end{aligned}
$$

Constants for Map joining

$$
\begin{aligned}
k_5 &= \frac{c_1(P-r)}{3(m-r)^2} \\
k_6 &= \left[\frac{c_1(3m-7r)}{6(m-1)}+\frac{c_2}{2}\right]\frac{(P-r)}{(m-r)^2} \\
k_7 &= \left[\frac{c_1(6r^2-6mr+m^2)}{6(m-r)}+\frac{c_2(m-2r)}{2}\right]\frac{(P-r)}{(m-r)^2} \\
k_8 &= c_3(P-r)
\end{aligned}
$$

$$
\begin{aligned}
k_{\ 9} &= -(2r+1) \\
k_{10} &= (1/2)(2Pr+P-r) \\
k_{11} &= (1/2)(2P^2r-3Pr^2+P^2-Pr+15r^3+ \\
&\quad +14r^2+2r) \\
k_{12} &= (1/2)(2P^2r-10Pr^2+3P^2r^2-12Pr^3- \\
&\quad -Pr-9r^4-12r^3-3r^2) \\
k_{13} &= (-1/2)(4P^2r^2-12Pr^3+3P^2r^3- \\
&\quad -9Pr^4+P^2r-3Pr^2) \\
k_{14} &= (-r) \\
k_{15} &= (1/6)(Pr+17r^2) \\
k_{16} &= (1/6)(3P^2r-8Pr^2-13r^3) \\
k_{17} &= (1/6)(2P^3r-9P^2r^2+13Pr^3) \\
k_{18} &= r \\
k_{19} &= 3r^2+3r \\
k_{20} &= 2r^3+2r^2
\end{aligned}
$$

ACKNOWLEDGMENT

This research has been funded in part by the Dirección General de Investigación of Spain under project DPI2003-07986.

REFERENCES

[1] J. A. Castellanos, J. M. M. Montiel, J. Neira, and J. D. Tardós, "The SPmap: A probabilistic framework for simultaneous localization and map building," *IEEE Trans. Robot. Automat.*, vol. 15, no. 5, pp. 948–953, 1999.

[2] J. Leonard, H. Durrant-Whyte, and I. Cox, "Dynamic map building for an autonomous mobile robot," *The International Journal of Robotics Research*, vol. 11, no. 4, pp. 286–298, 1992.

[3] S. Thrun, Y. Liu, D. Koller, A. Y. Ng, Z. Ghahramani, and H. Durrant-Whyte, "Simultaneous Localization and Mapping with Sparse Extended Information Filters," *The International Journal of Robotics Research*, vol. 23, no. 7-8, pp. 693–716, 2004.

[4] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit, "FastSLAM: A factored solution to the simultaneous localization and mapping problem," in *Proceedings of the AAAI National Conference on Artificial Intelligence*. Edmonton, Canada: AAAI, 2002.

[5] S. Thrun, "Particle filters in robotics," in *Proceedings of Uncertainty in AI (UAI)*, 2002.

[6] J. Castellanos, J. Neira, and J. Tardós, "Limits to the consistency of EKF-based SLAM," in *5th IFAC Symposium on Intelligent Autonomous Vehicles*, Lisbon, Portugal, 2004.

[7] J. Knight, A. Davison, and I. Reid, "Towards constant time SLAM using postponement," in *IEEE/RSJ Int'l Conf on Intelligent Robots and Systems*, Maui, Hawaii, 2001, pp. 406–412.

[8] J. E. Guivant and E. M. Nebot, "Optimization of the simultaneous localization and map-building algorithm for real-time implementation," *IEEE Trans. on Robotics and Automation*, vol. 17, no. 3, pp. 242–257, 2001.

[9] S. B. Williams, G. Dissanayake, and H. Durrant-Whyte, "An efficient approach to the simultaneous localisation and mapping problem," in *IEEE Int. Conf. on Robotics and Automation, ICRA*, vol. 1, Washington DC, 2002, pp. 406–411.

[10] J. E. Guivant and E. M. Nebot, "Solving computational and memory requirements of feature-based simultaneous localization and mapping algorithm," *IEEE Trans. Robot. Automat.*, vol. 19, no. 4, pp. 749–755, August 2003.

[11] J. Leonard and P. Newman, "Consistent, convergent and constant-time SLAM," in *Int. Joint Conf. on Artificial Intelligence*, Acapulco, Mexico, August 2003.

[12] M. Bosse, P. Newman, J. Leonard, and S. Telle, "Slam in large-scale cyclic environments using the atlas framework," *International Journal Of Robotics Research*, vol. 23, no. 12, pp. 1113–1139, December 2004.

[13] C. Estrada, J. Neira, and J. D. Tardós, "Hierarchical SLAM: real-time accurate mapping of large environments," *IEEE Trans. on Robotics*, 2005, to appear.

[14] J. Tardós, J. Neira, P. Newman, and J. Leonard, "Robust mapping and localization in indoor environments using sonar data," *Int. J. Robotics Research*, vol. 21, no. 4, pp. 311–330, 2002.