Motion Tracking with Fixed-lag Smoothing: Algorithm and Consistency Analysis

Tue-Cuong Dong-Si and Anastasios I. Mourikis
Dept. of Electrical Engineering, University of California, Riverside
E-mail: tdongsi@ee.ucr.edu, mourikis@ee.ucr.edu

***Abstract*— This paper presents a fixed-lag smoothing algorithm for tracking the motion of a mobile robot in real time. The algorithm processes measurements from proprioceptive (e.g., odometry, inertial measurement unit) and exteroceptive (e.g., camera, laser scanner) sensors, in order to estimate the trajectory of the vehicle. Smoothing is carried out in the information-filtering framework, and utilizes iterative minimization, which renders the method well-suited for applications where the effects of the measurements' nonlinearity are significant. The algorithm attains bounded computational complexity by marginalizing out older states. The key contribution of this work is a detailed analysis of the effects of the marginalization process on the consistency properties of the estimator. Based on this analysis, a linearization scheme that results in substantially improved accuracy, compared to the standard linearization approach, is proposed. Both simulation and real-world experimental results are presented, which demonstrate that the proposed method attains localization accuracy superior to that of competing approaches.**

I. Introduction and Relation to Prior Work

In this work, we focus on the problem of *motion tracking* for autonomous vehicle navigation. The simplest solution to this problem is *dead reckoning*, which consists of simply integrating proprioceptive (e.g., odometry) measurements, to obtain estimates of the vehicle's position and orientation. This process is straightforward, but results in rapid error accumulation, and leads to unacceptably large estimation uncertainty in most cases. To improve the accuracy of the estimates, additional sensory information is needed. This can be obtained from an exteroceptive sensor (e.g., a camera) that tracks static features in the environment to infer the vehicle's ego-motion. The task of fusing proprioceptive measurements and observations of local features for motion estimation is the focus of our work. We note that we are only interested in processing local motion information, i.e., we do not address loop closing or localization using *a priori* known landmarks.

Several approaches have been proposed for the task of motion tracking using feature observations. The simplest (and most computationally efficient) among these use the measurements to derive probabilistic constraints between *pairs* of consecutive poses (e.g., [1]–[5] and references therein). However, when a feature is observed multiple times, we can derive constraints involving *all* the poses from which the feature is seen [6]. Thus, in the (common) case where features are detected multiple times, pairwise processing is not appropriate. Instead, we must maintain estimates of a *sliding window* of poses, to be able to process all the relative-motion information provided by the feature measurements. Methods that estimate the state in a sliding window of time are commonly called *fixed-lag smoothing* algorithms [7].

While extended Kalman filtering (EKF)-based fixed-lag smoothing approaches to motion estimation have appeared in the literature (e.g., [6]), these are susceptible to a gradual buildup of linearization errors. This issue is especially significant in the context of motion tracking, where the absence of landmarks with *a priori* known positions means that errors continuously accumulate. For this reason a number of techniques have been proposed, primarily in the computer-vision literature, that iteratively re-linearize the measurements to better treat nonlinearity [8]–[12]. In all these approaches a sliding window of active states is maintained, comprising both camera poses and features, and iterative minimization is employed for obtaining estimates of all the currently active state variables. As the camera moves in space new states are being added, while old ones are discarded. A common limitation of the aforementioned techniques is, however, that the way in which older states are discarded is not optimal. Specifically, these are assumed to be fixed, and are used to "bootstrap" the trajectory estimates. This approximate approach ignores the uncertainty of the discared states, and results in suboptimal estimation.

The theoretically sound method for discarding older states from the sliding window is the process of *marginalization* [13]–[15]. This process, which appropriately accounts for the uncertainty of the older poses, is employed in our work. Specifically, the main contributions of this work are the following:

1) We describe in detail the derivation of the marginalization equations, and additionally, we present an analysis of the effects of the marginalization on the consistency of the state estimates. In particular, we show that due to the marginalization process, two different estimates of the same states are used in computing certain Jacobian matrices in the estimator. In turn, this is shown to cause an infusion of information along directions of the state space where no actual information is provided by the measurements (the unobservable directions of the state space). This "artificial" information causes the estimates to become *inconsistent* over time, i.e., it causes the actual error covariance to be larger than that reported by the estimator [16].

2) Based on our analysis, we propose a simple modification in the choice of linearization points, which prevents the introduction of artificial information in the estimator. The resulting algorithm is shown, through both simulation results

and real-world experiments, to perform better than competing approaches. In fact, our results show that the attained accuracy is almost indistinguishable to that of the full-state maximum-a-posteriori (MAP) estimator, for the cases examined. This shows that in choosing the linearization points one has to take into account the observability properties of the system at hand, to ensure that the linearized system shares the same properties.

We note that fixed-lag information smoothing algorithms have also appeared in [14], [15]. However, the effects of the marginalization process on the consistency of the estimates are not discussed in these publications. To the best of our knowledge, this is the first work to address this issue. The effects of the choice of linearization points on the consistency of an estimator have been explored before, but only in the context of EKF-based estimators (see, e.g., [17]).

II. Fixed-lag smoothing

A. Full-state MAP estimation

We begin by first discussing the full-state maximum-a-posteriori (MAP) estimator [18], which serves as the basis of the fixed-lag smoothing algorithm, and subsequently describe the marginalization process. This section also introduces the notation that will be used throughout the paper.

At time-step k, the full-state MAP estimator simultaneously estimates the entire history of robot poses, $\mathbf{r}_{0:k} = \{\mathbf{r}_0, \ldots, \mathbf{r}_k\}$, as well as the positions of all features observed by the robot, $\mathbf{l}_{1:n} = \{\mathbf{l}_1, \ldots, \mathbf{l}_n\}$. We denote the dimension of each robot pose and each landmark position by d_r and d_ℓ, respectively. Three sources of information are available to the estimator: (i) The prior information for the initial robot pose, described by a Gaussian pdf with mean $\hat{\mathbf{r}}_{p_0}$ and covariance matrix $\mathbf{R}_p(0)$. (ii) The robot motion model, described by: $\mathbf{r}_{i+1} = \mathbf{f}(\mathbf{r}_i, \mathbf{u}_i) + \mathbf{w}_i$, where $\mathbf{u}_i$ is the measured control input (e.g., odometry), and $\mathbf{w}_i$ is the process noise[1], assumed to be zero mean, white, and Gaussian, with covariance matrix $\mathbf{Q}_i$. (iii) Finally, the third source of information are the robot-to-landmark measurements (e.g., camera observations), described by: $\mathbf{z}_{ij} = \mathbf{h}(\mathbf{r}_i, \mathbf{l}_j) + \mathbf{n}_{ij}$, where $\mathbf{n}_{ij}$ is the measurement noise vector, assumed to be zero-mean, white, and Gaussian, with covariance matrix $\mathbf{R}_{ij}$.

The MAP estimator computes the state estimates that maximize the posterior pdf:

$$p(\mathbf{r}_{0:k}, \mathbf{l}_{1:n} | \mathbf{z}_{0:k}) = p(\mathbf{r}_0) \prod_{(i,j)\in\mathcal{S}_a(k)} p(\mathbf{z}_{ij}|\mathbf{r}_i, \mathbf{l}_j) \prod_{i=0}^{k-1} p(\mathbf{r}_{i+1}|\mathbf{r}_i, \mathbf{u}_i)$$

where the set $\mathcal{S}_a(k)$ contains the pairs of indices (i, j) that describe all the robot-to-landmark measurements through time k. Maximizing the posterior is equivalent to minimizing the cost function:

$$c(\mathbf{x}_k) = \frac{1}{2}||\mathbf{r}_0 - \hat{\mathbf{r}}_{p_0}||_{\mathbf{R}_p(0)} + \frac{1}{2} \sum_{(i,j)\in\mathcal{S}_a(k)} \gamma_{h_{ij}} + \frac{1}{2} \sum_{i=0}^{k-1} \gamma_{f_i} \tag{1}$$

[1]In the most general case, the process noise may appear in the motion model nonlinearly: $\mathbf{r}_{i+1} = \mathbf{f}(\mathbf{r}_i, \mathbf{u}_i, \mathbf{w}_i)$. The treatment of this more general case proceeds analogously, by linearization of the motion model with respect to the noise. We here present the additive-noise case for clarity.

where $\mathbf{x}_k$ denotes the vector containing all the estimated states, (i.e., all the states in $\{\mathbf{r}_{0:k}, \mathbf{l}_{1:n}\}$), and

$$\gamma_{h_{ij}} = ||\mathbf{z}_{ij} - \mathbf{h}(\mathbf{r}_i, \mathbf{l}_j)||_{\mathbf{R}_{ij}}, \quad \gamma_{f_i} = ||\mathbf{r}_{i+1} - \mathbf{f}(\mathbf{r}_i, \mathbf{u}_i)||_{\mathbf{Q}_i}$$

with the notation $||\mathbf{a}||_{\mathbf{M}} = \mathbf{a}^T\mathbf{M}^{-1}\mathbf{a}$.

$c(\mathbf{x}_k)$ is a nonlinear cost function, which can be minimized using iterative Gauss-Newton minimization [13]. At the ℓ-th iteration of this method, a correction, $\Delta\mathbf{x}^{(\ell)}$, to the current estimate, $\mathbf{x}_k^{(\ell)}$, is computed by minimizing the second-order Taylor-series approximation of the cost function:

$$c(\mathbf{x}_k^{(\ell)} + \Delta\mathbf{x}) \simeq c(\mathbf{x}^{(\ell)}) + \mathbf{b}^{(\ell)T}\Delta\mathbf{x} + \frac{1}{2}\Delta\mathbf{x}^T\mathbf{A}^{(\ell)}\Delta\mathbf{x} \tag{2}$$

where $\mathbf{b}^{(\ell)} = \nabla c(\mathbf{x}_k^{(\ell)})$ and $\mathbf{A}^{(\ell)} = \nabla^2 c(\mathbf{x}_k^{(\ell)})$ denote the gradient and Hessian of c with respect to $\mathbf{x}_k$, evaluated at $\mathbf{x}_k^{(\ell)}$. Specifically, $\mathbf{b}^{(\ell)}$ is given by

$$\mathbf{b}^{(\ell)} = \mathbf{b}_0^p(\mathbf{x}_k^{(\ell)}) + \mathbf{b}_{\mathcal{S}_a(k)}^h(\mathbf{x}_k^{(\ell)}) + \mathbf{b}_{0:k}^f(\mathbf{x}_k^{(\ell)}) \tag{3}$$

where the three terms are due to the prior at time-step 0, the robot-to-landmark measurements indexed by $\mathcal{S}_a(k)$, and the odometry measurements in the time interval $[0, k-1]$, respectively:

$$\mathbf{b}_0^p(\mathbf{x}_k^{(\ell)}) = \mathbf{\Pi}_{d_r}^T\mathbf{A}_p(0)(\mathbf{r}_0^{(\ell)} - \hat{\mathbf{r}}_{p_0}), \tag{4}$$

$$\mathbf{b}_{\mathcal{S}_a(k)}^h(\mathbf{x}_k^{(\ell)}) = -\sum_{(i,j)\in\mathcal{S}_a(k)} \mathbf{H}_{ij}^{(\ell)T}\mathbf{R}_{ij}^{-1}\left(\mathbf{z}_{ij} - \mathbf{h}(\mathbf{r}_i^{(\ell)}, \mathbf{l}_j^{(\ell)})\right) \tag{5}$$

$$\mathbf{b}_{0:k}^f(\mathbf{x}_k^{(\ell)}) = \sum_{i=0}^{k-1} \mathbf{G}_i^{(\ell)T}\mathbf{Q}_i^{-1}\left(\mathbf{r}_{i+1}^{(\ell)} - \mathbf{f}(\mathbf{r}_i^{(\ell)}, \mathbf{u}_i)\right) \tag{6}$$

In these expressions we denote $\mathbf{A}_p(0) = \mathbf{R}_p(0)^{-1}$, and $\mathbf{\Pi}_{d_r} = \begin{bmatrix}\mathbf{I}_{d_r} & \mathbf{0} & \mathbf{0} & \ldots\end{bmatrix}$, with $\mathbf{I}_{d_r}$ being the $d_r \times d_r$ identity matrix. The matrices $\mathbf{H}_{ij}^{(\ell)}$ and $\mathbf{G}_i^{(\ell)}$ are the Jacobians of the measurement function, $\mathbf{h}(\mathbf{r}_i, \mathbf{l}_j)$, and of the function $\mathbf{g}_i = \mathbf{r}_{i+1} - \mathbf{f}(\mathbf{r}_i, \mathbf{u}_i)$, with respect to $\mathbf{x}_k$, evaluated at $\mathbf{x}_k^{(\ell)}$. Since both the measurement function and the motion model involve only two states (either one robot pose and one landmark, or two consecutive robot poses), the structure of both $\mathbf{H}_{ij}^{(\ell)}$ and $\mathbf{G}_i^{(\ell)}$ is very sparse. In particular,

$$\mathbf{H}_{ij}^{(\ell)} = \begin{bmatrix}\mathbf{0} & \ldots & \mathbf{H}_{R_{ij}}^{(\ell)} & \ldots & \mathbf{H}_{L_{ij}}^{(\ell)} & \ldots & \mathbf{0}\end{bmatrix} \tag{7}$$

where $\mathbf{H}_{R_{ij}}^{(\ell)}$ and $\mathbf{H}_{L_{ij}}^{(\ell)}$ are the Jacobians of $\mathbf{h}$ with respect to the robot pose and the landmark position, respectively, and

$$\mathbf{G}_i^{(\ell)} = \begin{bmatrix}\mathbf{0} & \ldots & -\mathbf{\Phi}_i^{(\ell)} & \mathbf{I}_{d_r} & \ldots & \mathbf{0}\end{bmatrix} \tag{8}$$

where $\mathbf{\Phi}_i^{(\ell)}$ is the Jacobian of $\mathbf{f}(\mathbf{r}_i, \mathbf{u}_i)$ with respect to $\mathbf{r}_i$.

In the Gauss-Newton method, and for small-residual problems, the Hessian matrix can be well approximated by [13]:

$$\mathbf{A}^{(\ell)} = \mathbf{\Lambda}_0^p + \mathbf{\Lambda}_{\mathcal{S}_a(k)}^h(\mathbf{x}_k^{(\ell)}) + \mathbf{\Lambda}_{0:k}^f(\mathbf{x}_k^{(\ell)}), \tag{9}$$

$$\mathbf{\Lambda}_0^p = \mathbf{\Pi}_{d_r}^T\mathbf{A}_p(0)\mathbf{\Pi}_{d_r} \tag{10}$$

$$\mathbf{\Lambda}_{\mathcal{S}_a(k)}^h(\mathbf{x}_k^{(\ell)}) = \sum_{(i,j)\in\mathcal{S}_a(k)} \mathbf{H}_{ij}^{(\ell)T}\mathbf{R}_{ij}^{-1}\mathbf{H}_{ij}^{(\ell)} \tag{11}$$

$$\mathbf{\Lambda}_{0:k}^f(\mathbf{x}_k^{(\ell)}) = \sum_{i=0}^{k-1} \mathbf{G}_i^{(\ell)T}\mathbf{Q}_i^{-1}\mathbf{G}_i^{(\ell)} \tag{12}$$

In the above notation, $\mathbf{\Lambda}_0^p$ represents the information due to the prior at time 0, $\mathbf{\Lambda}_{\mathcal{S}_a(k)}^h(\mathbf{x})$ represents the information matrix due to the measurements indexed by $\mathcal{S}_a(k)$, evaluated using linearization about $\mathbf{x}$, and $\mathbf{\Lambda}_{0:k}^f(\mathbf{x})$ is the information matrix due to the process model for the time interval $[0, k]$, evaluated using linearization about $\mathbf{x}$.

The value of $\Delta\mathbf{x}^{(\ell)}$ minimizing the cost function (2) is found by solving the linear system:

$$\mathbf{A}^{(\ell)}\Delta\mathbf{x}^{(\ell)} = -\mathbf{b}^{(\ell)} \tag{13}$$

Due to the sparse structure of $\mathbf{H}_{ij}^{(\ell)}$ and $\mathbf{G}_i^{(\ell)}$, the Hessian matrix $\mathbf{A}^{(\ell)}$ is sparse (see Fig.1). This can be exploited to speed-up the solution of the above linear system [13], [18].

B. Marginalization of old states

As the robot continuously moves and observes new features, the size of the state vector $\mathbf{x}_k$ constantly increases (approximately linearly in time, if the density of features is constant). Therefore, in order to obtain an algorithm with bounded computational complexity, suitable for real-time applications, we resort to marginalization of older poses. In this section, we derive the marginalization equations from the perspective of minimization of the cost function c.

We consider the following scenario: The robot collects measurements during the time interval $[0, k]$, and full-state MAP estimation is carried out at time-step k. Then, the states $\mathbf{x}_m = \{\mathbf{r}_0, \ldots, \mathbf{r}_{m-1}, \mathbf{l}_1, \ldots, \mathbf{l}_{m_l}\}$ (i.e., the m oldest robot poses and the m_l oldest landmarks, which we can no longer observe) are removed (i.e., marginalized out), and only the states $\mathbf{x}_r = \{\mathbf{r}_m, \ldots, \mathbf{r}_k, \mathbf{l}_{m_l+1}, \ldots, \mathbf{l}_n\}$ remain active in the sliding window. The robot keeps moving and collecting measurements in the time interval $[k+1, k']$, and as a result, the history of states is augmented by the new robot and landmark states $\mathbf{x}_n = \{\mathbf{r}_{k+1}, \ldots, \mathbf{r}_{k'}, \mathbf{l}_{n+1}, \ldots, \mathbf{l}_{n'}\}$. Now, at time-step k', the sliding window contains the states $\mathbf{x}_r$ and $\mathbf{x}_n$, and we would like to compute the MAP estimate for these states.

To compute the optimal MAP estimate at time k', one has to minimize a cost function $c(\mathbf{x}_{k'})$, which is analogous to the one in (1). This cost function has a special structure:

$$c(\mathbf{x}_{k'}) = c(\mathbf{x}_m, \mathbf{x}_r, \mathbf{x}_n) = c_n(\mathbf{x}_r, \mathbf{x}_n) + c_m(\mathbf{x}_r, \mathbf{x}_m) \tag{14}$$

The term $c_n(\mathbf{x}_r, \mathbf{x}_n)$ in the above expression contains all quadratic terms that involve states in $\mathbf{x}_r$ only, states in $\mathbf{x}_n$ only, or terms involving one state in $\mathbf{x}_r$ and one in $\mathbf{x}_n$. On the other hand, $c_m(\mathbf{x}_r, \mathbf{x}_m)$ contains all quadratic terms that involve states in $\mathbf{x}_m$ only, as well as terms involving one state in $\mathbf{x}_m$ and one in $\mathbf{x}_r$. Since states marginalized at time-step k do not participate in any measurements after that time (they are older robot poses, and features that can no longer be seen), there do not exist terms jointly involving states in $\mathbf{x}_n$ and $\mathbf{x}_m$.

It is important to observe that even though the states $\mathbf{x}_m$ and the associated measurements are no longer kept in the estimator, their exact values are *not* known, and therefore the above minimization needs to be performed with respect to $\mathbf{x}_m$ as well. In what follows, we show what data we need to keep after the marginalization at time-step k, to be able to carry out this minimization. We start by utilizing the decomposition of the cost function in (14), to employ the following property, which holds for any multi-variable optimization problem:

$$\begin{aligned}\min_{\mathbf{x}_m,\mathbf{x}_r,\mathbf{x}_n} c(\mathbf{x}_m, \mathbf{x}_r, \mathbf{x}_n) &= \min_{\mathbf{x}_r,\mathbf{x}_n}\left(\min_{\mathbf{x}_m} c(\mathbf{x}_m, \mathbf{x}_r, \mathbf{x}_n)\right) \\ &= \min_{\mathbf{x}_r,\mathbf{x}_n}\left(c_n(\mathbf{x}_r, \mathbf{x}_n) + \min_{\mathbf{x}_m} c_m(\mathbf{x}_r, \mathbf{x}_m)\right)\end{aligned} \tag{15}$$

The above reformulation of the minimization problem entails no approximation. We now focus on the minimization of c_m with respect to $\mathbf{x}_m$. c_m is given by:

$$c_m(\mathbf{x}_r, \mathbf{x}_m) = \frac{1}{2}||\mathbf{r}_0 - \hat{\mathbf{r}}_{p_0}||_{\mathbf{R}_p(0)} + \frac{1}{2}\sum_{(i,j)\in\mathcal{S}_m}\gamma_{h_{ij}} + \frac{1}{2}\sum_{i=0}^{m-1}\gamma_{f_i}$$

where $\mathcal{S}_m$ is the set of indices (i, j) describing all the robot-to-landmark measurements that involve either marginalized robot poses or marginalized landmarks (or both). Since the measurement and process-model functions are nonlinear, the minimization of c_m with respect to $\mathbf{x}_m$ cannot be carried out analytically, and we once again employ the second-order Taylor-series approximation:

$$\begin{aligned}c_m \simeq\ & c_m(\hat{\mathbf{x}}_r(k), \hat{\mathbf{x}}_m(k)) + \mathbf{b}_m(k)^T \begin{bmatrix}\mathbf{x}_m - \hat{\mathbf{x}}_m(k) \\ \mathbf{x}_r - \hat{\mathbf{x}}_r(k)\end{bmatrix} \\ & + \frac{1}{2}\begin{bmatrix}\mathbf{x}_m - \hat{\mathbf{x}}_m(k) \\ \mathbf{x}_r - \hat{\mathbf{x}}_r(k)\end{bmatrix}^T \mathbf{A}_m(k) \begin{bmatrix}\mathbf{x}_m - \hat{\mathbf{x}}_m(k) \\ \mathbf{x}_r - \hat{\mathbf{x}}_r(k)\end{bmatrix}\end{aligned} \tag{16}$$

where $\mathbf{b}_m(k) = \nabla c_m(\hat{\mathbf{x}}_r(k), \hat{\mathbf{x}}_m(k))$ is the gradient, and $\mathbf{A}_m(k) = \nabla^2 c_m(\hat{\mathbf{x}}_r(k), \hat{\mathbf{x}}_m(k))$ the Hessian matrix of c_m, evaluated at the MAP estimates of $\mathbf{x}_r$ and $\mathbf{x}_m$ at time-step k:

$$\begin{aligned}\mathbf{b}_m(k) =\ & \mathbf{b}_0^p(\hat{\mathbf{x}}_r(k), \hat{\mathbf{x}}_m(k)) + \mathbf{b}_{\mathcal{S}_m}^h(\hat{\mathbf{x}}_r(k), \hat{\mathbf{x}}_m(k)) \\ & + \mathbf{b}_{0:m}^f(\hat{\mathbf{x}}_r(k), \hat{\mathbf{x}}_m(k))\end{aligned} \tag{17}$$

$$\mathbf{A}_m(k) = \mathbf{\Lambda}_0^p + \mathbf{\Lambda}_{\mathcal{S}_m}^h(\hat{\mathbf{x}}_r(k), \hat{\mathbf{x}}_m(k)) + \mathbf{\Lambda}_{0:m}^f(\hat{\mathbf{x}}_r(k), \hat{\mathbf{x}}_m(k)) \tag{18}$$

For the following derivations, it will be convenient to define the block partitioning of the gradient and Hessian of c_m as follows (see Fig. 1):

$$\mathbf{b}_m(k) = \begin{bmatrix}\mathbf{b}_{mm}(k) \\ \mathbf{b}_{mr}(k)\end{bmatrix} \quad \mathbf{A}_m(k) = \begin{bmatrix}\mathbf{A}_{mm}(k) & \mathbf{A}_{mr}(k) \\ \mathbf{A}_{rm}(k) & \mathbf{A}_{rr}(k)\end{bmatrix} \tag{19}$$

where the dimensions of the blocks correspond to the dimensions of $\mathbf{x}_m$ and $\mathbf{x}_r$, and the time-step argument (k) denotes the fact that all quantities in these matrices are evaluated using the state estimates at time k. At this point, we note that the cost function in (16) is a quadratic function of $\mathbf{x}_m$, and its minimum with respect to $\mathbf{x}_m$ is

$$\min_{\mathbf{x}_m} c_m \simeq \kappa + \mathbf{b}_p^T(k)(\mathbf{x}_r - \hat{\mathbf{x}}_r(k)) + \frac{1}{2}||\mathbf{x}_r - \hat{\mathbf{x}}_r(k)||_{\mathbf{A}_p(k)}$$

where κ is a constant independent of $\mathbf{x}_r$ and $\mathbf{x}_m$, and

$$\mathbf{b}_p(k) = \mathbf{b}_{mr}(k) - \mathbf{A}_{rm}(k)\mathbf{A}_{mm}(k)^{-1}\mathbf{b}_{mm}(k) \tag{20}$$

$$\mathbf{A}_p(k) = \mathbf{A}_{rr}(k) - \mathbf{A}_{rm}(k)\mathbf{A}_{mm}(k)^{-1}\mathbf{A}_{mr}(k) \tag{21}$$

Combining this result with that of (15), we see that the minimization of the cost function $c(\mathbf{x}_m, \mathbf{x}_r, \mathbf{x}_n)$ with respect

to the entire history of states is approximately equivalent to the minimization, with respect to $\{\mathbf{x}_r, \mathbf{x}_n\}$, of:

$$c'(\mathbf{x}_r, \mathbf{x}_n) = \mathbf{b}_p^T(k)(\mathbf{x}_r - \hat{\mathbf{x}}_r(k)) + \frac{1}{2}||\mathbf{x}_r - \hat{\mathbf{x}}_r(k)||_{\mathbf{A}_p(k)} + \frac{1}{2}\sum_{(i,j)\in\mathcal{S}_a(k')} \gamma_{h_{ij}} + \frac{1}{2}\sum_{i=m}^{k'-1} \gamma_{f_i} \quad (22)$$

where the set $\mathcal{S}_a(k')$ contains the (i, j) indices corresponding to all the active measurements at time-step k' (i.e., all measurements involving states in $\mathbf{x}_r$ and $\mathbf{x}_n$). It is important to note that the above cost function does *not* depend on $\mathbf{x}_m$. Thus, if after marginalization at time-step k we store $\mathbf{A}_p(k)$, $\mathbf{b}_p(k)$, and $\hat{\mathbf{x}}_r(k)$, the above minimization can still be carried out. The *only* approximation here lies in the fact that the term c_m has been permanently approximated by the second-order Taylor series approximation of (16). This will introduce small errors in the MAP estimates for $\{\mathbf{x}_r, \mathbf{x}_n\}$, but if the states we chose to marginalize at time-step k are old, "mature" ones, for which good estimates are available, the effect of the approximation will be small. On the other hand, the gain from employing this approximation is that the states $\mathbf{x}_m$ and all measurements that directly involve them can be discarded, yielding an algorithm with constant-time and constant-memory requirements.

The minimization of $c'(\mathbf{x}_r, \mathbf{x}_n)$ at time-step k' is carried out by the Gauss-Newton method. Similarly to the previous case, during the ℓ-th iteration, the correction to the active states $\{\mathbf{x}_r, \mathbf{x}_n\}$, is computed by solving the sparse linear system $\mathbf{A}^{(\ell)}\Delta\mathbf{x} = -\mathbf{b}^{(\ell)}$, where:

$$\mathbf{b}^{(\ell)} = \mathbf{b}_k^p(\mathbf{x}_r^{(\ell)}) + \mathbf{b}_{\mathcal{S}_a(k')}^h(\mathbf{x}_r^{(\ell)}, \mathbf{x}_n^{(\ell)}) + \mathbf{b}_{m:k'}^f(\mathbf{x}_r^{(\ell)}, \mathbf{x}_n^{(\ell)}) \quad (23)$$

$$\mathbf{A}^{(\ell)} = \mathbf{\Lambda}_k^p + \mathbf{\Lambda}_{\mathcal{S}_a(k')}^h(\mathbf{x}_r^{(\ell)}, \mathbf{x}_n^{(\ell)}) + \mathbf{\Lambda}_{m:k'}^f(\mathbf{x}_r^{(\ell)}, \mathbf{x}_n^{(\ell)}) \quad (24)$$

The terms in the above expressions are defined analogously to those in (3)-(9), with the exception of $\mathbf{b}_k^p(\mathbf{x}_r^{(\ell)})$, which is given by:

$$\mathbf{b}_k^p(\mathbf{x}_r^{(\ell)}) = \mathbf{\Pi}_r\mathbf{b}_p(k) + \mathbf{\Pi}_r\mathbf{A}_p(k)(\mathbf{x}_r^{(\ell)} - \hat{\mathbf{x}}_r(k)) \quad (25)$$

with $\mathbf{\Pi}_r = [\mathbf{I}_{\dim\mathbf{x}_r} \quad \mathbf{0} \quad \mathbf{0} \quad ...]$.

C. Fixed-lag smoothing algorithm

We can now describe the entire fixed-lag smoothing estimation algorithm (see Algorithm 1). During each estimation step, updates are computed for all the states that are in the currently active state vector by solving the sparse linear system defined by (23) and (24). If the covariance matrix of the sliding-window estimates is needed, it can be computed after the iteration converges as the inverse of the information matrix $\mathbf{A}^{(\ell)}$. Moreover, if desired we can marginalize out a number of states, so as to reduce the size of the actively estimated state vector. If, at time-step k, we choose to marginalize out the states $\mathbf{x}_m$ and keep only $\mathbf{x}_r$, the current estimate $\hat{\mathbf{x}}_r(k)$ will take the place of the prior, the prior information matrix $\mathbf{A}_p(k)$ will be computed via (21), and the vector $\mathbf{b}_p(k)$ via (20). Once these quantities have been computed, the states $\mathbf{x}_m$ and all measurements that directly involve them can be discarded.

Algorithm 1 Fixed-Lag Information Smoothing

Initialization:

Prior information: $\mathbf{A}_p(0) = \mathbf{R}_p^{-1}(0)$, Prior estimate: $\hat{\mathbf{x}}_p = \hat{\mathbf{r}}_{p_0}$, Prior constant vector for use in (25): $\mathbf{b}_p = \mathbf{0}$.

MAP Estimation

Starting from an initial estimate, iteratively compute corrections to the state by solving the sparse linear system $\mathbf{A}^{(\ell)}\Delta\mathbf{x} = -\mathbf{b}^{(\ell)}$, where $\mathbf{A}^{(\ell)}$ and $\mathbf{b}^{(\ell)}$ are defined in (23)-(25), respectively. Repeat until convergence.

Marginalization

- Set the new prior estimate as $\hat{\mathbf{x}}_p = \hat{\mathbf{x}}_r(k)$, and compute $\mathbf{A}_p$ and $\mathbf{b}_p$ via (20) and (21).
- Remove the states $\mathbf{x}_m$ from the active state vector, and discard all measurements that involve these states.

It is important to examine the structure of the prior information matrix $\mathbf{A}_p$. As discussed earlier, in the case of the full-state MAP, the Hessian matrix $\mathbf{A}$ is sparse, which significantly speeds up computation. We now show that, typically, the same holds in the case of fixed-lag smoothing. Fig. 1 illustrates the sparsity pattern of the full-state Hessian, $\mathbf{A}$, for a typical trajectory. We observe that in the case of motion tracking, when (i) no loop closures occur, (ii) the number of landmarks seen by the robot at any time instant is limited, and (iii) features are not tracked for very long time periods, then $\mathbf{A}$ is a sparse band matrix. Moreover, it is well-known that the marginalization of a state variable only introduces fill-in in rows and columns corresponding to variables directly connected to it via measurements [13]. Therefore, the marginalization of old robot poses and old landmarks typically results in a *sparse* matrix $\mathbf{A}_p$, as shown in Fig. 1. Thus we are still able to employ sparse-matrix techniques for the solution of the Gauss-Newton system.

III. Estimator consistency

A key benefit of the fixed-lag smoother presented in the preceding section is the fact that it iteratively re-linearizes the measurements. This enables the algorithm to reduce the effects of the linearization errors, and thus to attain improved accuracy, compared to methods which do not employ re-linearization (e.g., EKF methods). Clearly, having small linearization errors is a key requirement for the consistency of an estimator; if large, unmodeled linearization errors exist, the accuracy claimed by the estimator will be too optimistic, and the estimates will be inconsistent. In this section, we examine a different factor that can cause inconsistency of the estimates. Specifically, we show that when the Jacobian matrices of the measurements are computed using the latest available state estimates (the "standard" option), the smoother will gain fictitious information, along directions of the state space where no real information is provided by the measurements. The immediate result of this is inconsistency, i.e., state estimates whose accuracy is worse than the one claimed by the estimator. Moreover, the over-confidence of

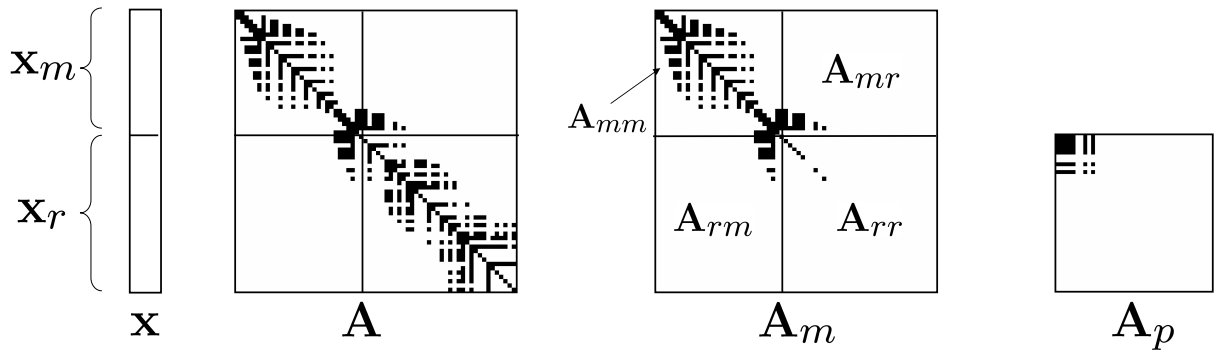


Fig. 1. Illustration of the sparsity patterns in the matrices appearing in the fixed-lag smoothing algorithm. Here we employ a temporal ordering of the variables (i.e., robot poses and landmarks enter the state vector in the order they appear in time), although during the solution of the system alternative variable ordering can be used if desired, to speed-up computations.

the estimator about the accuracy of the state estimates along certain directions of the state space leads to inaccurate state updates, and thus a degradation of accuracy, as shown in the results of the next section.

In what follows, we focus on the scenario described in Section II-B, i.e., marginalization of the states $\mathbf{x}_m$ at time-step k, and a new estimation step at time-step k'. Once the Gauss-Newton iteration at time k' has converged, the information (i.e., inverse covariance) matrix for the active states is given by

$$\begin{aligned}\mathbf{A}(k') = \mathbf{\Lambda}_k^p + \mathbf{\Lambda}^h_{\mathcal{S}_a(k')}\big(\hat{\mathbf{x}}_r(k'), \hat{\mathbf{x}}_n(k')\big) \\ + \mathbf{\Lambda}^f_{m:k'}\big(\hat{\mathbf{x}}_r(k'), \hat{\mathbf{x}}_n(k')\big)\end{aligned} \tag{26}$$

Our goal is to show that the linearization employed during the marginalization process results in the addition of non-existent information to the estimator, by studying the properties of $\mathbf{A}(k')$. To this end, we first note that $\mathbf{A}(k')$ is the Schur complement of $\mathbf{A}_{mm}(k)$ in the following matrix:

$$\begin{aligned}\mathbf{A}^{\text{mar}}_{\text{full}}(k') &= \mathbf{\Lambda}_0^p + \mathbf{\Lambda}^h_{\mathcal{S}_m}\big(\hat{\mathbf{x}}_m(k), \hat{\mathbf{x}}_r(k)\big) + \mathbf{\Lambda}^f_{0:m}\big(\hat{\mathbf{x}}_m(k), \hat{\mathbf{x}}_r(k)\big) \\ &\quad + \mathbf{\Lambda}^h_{\mathcal{S}_a(k')}\big(\hat{\mathbf{x}}_r(k'), \hat{\mathbf{x}}_n(k')\big) + \mathbf{\Lambda}^f_{m:k'}\big(\hat{\mathbf{x}}_r(k'), \hat{\mathbf{x}}_n(k')\big) \\ &= \begin{bmatrix} \mathbf{A}_{mm}(k) & \mathbf{A}_{mr}(k)\mathbf{\Pi}_r \\ \mathbf{\Pi}_r^T\mathbf{A}_{rm}(k) & \mathbf{\Pi}_r^T\mathbf{A}_{rr}(k)\mathbf{\Pi}_r \end{bmatrix} + \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{A}(k') - \mathbf{\Lambda}_k^p \end{bmatrix}\end{aligned} \tag{27}$$

It is important to observe that $\mathbf{A}^{\text{mar}}_{\text{full}}(k')$ represents the available information for the entire history of states $\{\mathbf{x}_m, \mathbf{x}_r, \mathbf{x}_n\}$. To see why this is the case, recall that the union of the sets $\mathcal{S}_m$ and $\mathcal{S}_a(k')$ is the set of all measurements recorded in the time interval $[0, k']$, and thus the matrices $\mathbf{\Lambda}^h_{\mathcal{S}_m}\big(\hat{\mathbf{x}}_m(k), \hat{\mathbf{x}}_r(k)\big)$ and $\mathbf{\Lambda}^h_{\mathcal{S}_a(k')}\big(\hat{\mathbf{x}}_r(k'), \hat{\mathbf{x}}_n(k')\big)$, taken together, represent all the available measurement information. Similarly, $\mathbf{\Lambda}^f_{0:m}\big(\hat{\mathbf{x}}_m(k), \hat{\mathbf{x}}_r(k)\big)$ and $\mathbf{\Lambda}^f_{m:k'}\big(\hat{\mathbf{x}}_r(k'), \hat{\mathbf{x}}_n(k')\big)$ together represent all the available process-model information, while $\mathbf{\Lambda}_0^p$ is the prior. Thus, the matrix $\mathbf{A}^{\text{mar}}_{\text{full}}(k')$ is analogous to the information matrix that would arise from a full-state MAP estimate at time-step k', but with the important difference that in $\mathbf{A}^{\text{mar}}_{\text{full}}(k')$, some of the Jacobian matrices are evaluated at the state estimates $\{\hat{\mathbf{x}}_m(k), \hat{\mathbf{x}}_r(k)\}$, while others are evaluated at $\{\hat{\mathbf{x}}_r(k'), \hat{\mathbf{x}}_n(k')\}$. We now show that this difference results in the "infusion" of artificial information into the estimation process.

To prove this result, we focus on the information provided to the estimation process by the process model and the feature measurements. For this reason, for the moment we consider the case where the prior is zero, i.e., $\mathbf{\Lambda}^p = 0$. In this case, $\mathbf{A}^{\text{mar}}_{\text{full}}(k')$ becomes

$$\begin{aligned}\mathbf{A}^{\text{mar}}_{\text{full}}(k') = \mathbf{\Lambda}^h_{\mathcal{S}_m}\big(\hat{\mathbf{x}}_m(k), \hat{\mathbf{x}}_r(k)\big) + \mathbf{\Lambda}^f_{0:m}\big(\hat{\mathbf{x}}_m(k), \hat{\mathbf{x}}_r(k)\big) \\ + \mathbf{\Lambda}^h_{\mathcal{S}_a(k')}\big(\hat{\mathbf{x}}_r(k'), \hat{\mathbf{x}}_n(k')\big) + \mathbf{\Lambda}^f_{m:k'}\big(\hat{\mathbf{x}}_r(k'), \hat{\mathbf{x}}_n(k')\big)\end{aligned} \tag{28}$$

On the other hand, if we had carried out a full-state MAP estimation at time step k', without having previously marginalized any states, the information matrix would be:

$$\begin{aligned}\mathbf{A}^{\text{nm}}_{\text{full}}(k') = \mathbf{\Lambda}^h_{\mathcal{S}_m \bigcup \mathcal{S}_a(k')}\big(\hat{\mathbf{x}}_m(k'), \hat{\mathbf{x}}_r(k'), \hat{\mathbf{x}}_n(k')\big) \\ + \mathbf{\Lambda}^f_{0:k'}\big(\hat{\mathbf{x}}_m(k'), \hat{\mathbf{x}}_r(k'), \hat{\mathbf{x}}_n(k')\big)\end{aligned} \tag{29}$$

A key result that we prove is that [19]

$$\text{rank}(\mathbf{A}^{\text{mar}}_{\text{full}}(k')) = \text{rank}(\mathbf{A}^{\text{nm}}_{\text{full}}(k')) + 1 \tag{30}$$

This shows that computing the information matrix $\mathbf{A}^{\text{mar}}_{\text{full}}(k')$ using two different estimates for the states $\mathbf{x}_r$ leads to an increase of its rank. Clearly, this increase is incorrect, since it is not justified by any new measurement information.

Due to limited space the full proof of the above result cannot be included here, and the interested reader is referred to [19] for the details of the proofs for both 2D and 3D motion estimation. Instead, we here provide some intuition behind the result. Specifically, for the case of 2D motion, in [19] we show that the matrix $\mathbf{A}^{\text{nm}}_{\text{full}}(k')$ has a nullspace of dimension three, spanned by the columns of the matrix:

$$\mathbf{N}\big(\hat{\mathbf{x}}_m(k'), \hat{\mathbf{x}}_r(k'), \hat{\mathbf{x}}_n(k')\big) = \begin{bmatrix} \mathbf{I}_2 & \mathbf{J}\hat{\mathbf{p}}_{R_0}(k') \\ \mathbf{0}_{1\times 2} & 1 \\ \vdots & \vdots \\ \mathbf{I}_2 & \mathbf{J}\hat{\mathbf{p}}_{R_{k'}}(k') \\ \mathbf{0}_{1\times 2} & 1 \\ \mathbf{I}_2 & \mathbf{J}\hat{\mathbf{p}}_{L_1}(k') \\ \vdots & \vdots \\ \mathbf{I}_2 & \mathbf{J}\hat{\mathbf{p}}_{L_{n'}}(k') \end{bmatrix} \tag{31}$$

where $\hat{\mathbf{p}}_{R_i}(k')$ and $\hat{\mathbf{p}}_{L_j}(k')$ are the estimates for the i-th robot position and j-th landmark position, respectively, computed at time-step k'. The physical interpretation of this result is that using only the robot-to-landmark measurements, only the *relative* positions between the robot poses and

the landmarks can be estimated with bounded uncertainty, while the global position and orientation of the state vector cannot be determined. (Note that the first two columns of $\mathbf{N}$ correspond to global translations of the state vector, while the third column to global rotations).

Turning our attention to the matrix $\mathbf{A}_{\text{full}}^{\text{mar}}(k')$ in (28), we again point out that in this matrix, two different estimates for $\mathbf{x}_r$ appear. These estimates are the ones computed at the time of marginalization, $\hat{\mathbf{x}}_r(k)$, and those computed after the Gauss-Newton iteration at time k', $\hat{\mathbf{x}}_r(k')$. This is the *only* difference between the matrices $\mathbf{A}_{\text{full}}^{\text{mar}}(k')$ and $\mathbf{A}_{\text{full}}^{\text{nm}}(k')$. Thus, $\mathbf{A}_{\text{full}}^{\text{mar}}(k')$ is essentially a perturbed version of $\mathbf{A}_{\text{full}}^{\text{nm}}(k')$, with the magnitude of the perturbation being proportional to $||\hat{\mathbf{x}}_r(k) - \hat{\mathbf{x}}_r(k')||$. This perturbation causes the third column of $\mathbf{N}(\hat{\mathbf{x}}_m(k'), \hat{\mathbf{x}}_r(k'), \hat{\mathbf{x}}_n(k'))$ not to belong to the nullspace of $\mathbf{A}_{\text{full}}^{\text{mar}}(k')$, and the dimension of this nullspace is only *two*, instead of three [19]. This means that the estimator acquires erroneous information about the global orientation, as a result of the use of two different linearization points for $\mathbf{x}_r$.

The result of (30) shows that the rank of the information matrix for the *entire* state history, $\mathbf{A}_{\text{full}}^{\text{mar}}(k')$, is incorrectly increased. Using (27) and the properties of the Schur complement, we can find a direct relationship between the rank of the information matrix for the entire state history and the information matrix for the currently active states, $\mathbf{A}(k')$:

$$\text{rank}(\mathbf{A}(k')) = \text{rank}(\mathbf{A}_{\text{full}}^{\text{mar}}(k')) - \text{rank}(\mathbf{A}_{mm}(k)) \quad (32)$$

Since $\mathbf{A}_{mm}(k)$ is in general a full-rank matrix, we conclude that the rank of the information matrix for the active states at time-step k' is increased by one, compared to its correct value.

We have thus shown that when no prior estimates are available, the marginalization process results in an erroneous increase in the rank of the state information matrix. Analogous conclusions can be drawn for the general situation, where prior information also exists. Specifically, in this case the rank of the information matrix is not increased (the matrix is already full-rank), but the "addition" of information in certain directions of the state space still happens. The immediate result of this is inconsistent estimates, i.e., estimates whose accuracy is worse than that claimed by the filter. Ultimately, this leads to a degradation of the accuracy of the state estimates themselves, as shown in the results presented in Section IV.

A. Improvement of the estimator's consistency

We now propose a simple solution to the problem of "creating" artificial information through the marginalization process. For this purpose, only a slight modification of the fixed-lag smoothing algorithm is needed. Specifically, in computing the Jacobians used in (24), we employ the *prior* estimates, rather than the *current* ones, for states for which a prior exists. Thus, (24) is changed to:

$$\mathbf{A}^{(\ell)} = \mathbf{\Lambda}_k^p + \mathbf{\Lambda}_{\mathcal{S}_a(k')}^h(\hat{\mathbf{x}}_r(k), \mathbf{x}_n^{(\ell)}) + \mathbf{\Lambda}_{m:k'}^f(\hat{\mathbf{x}}_r(k), \mathbf{x}_n^{(\ell)})$$

In the above the only estimate of $\mathbf{x}_r$ used is $\hat{\mathbf{x}}_r(k)$. In [19] we show that as a result, the rank of the matrix $\mathbf{A}_{\text{full}}^{\text{mar}}(k')$ does not increase when marginalization takes place (the nullspace of this matrix is then spanned by the columns of $\mathbf{N}(\hat{\mathbf{x}}_m(k), \hat{\mathbf{x}}_r(k), \hat{\mathbf{x}}_n(k'))$, and the influx of invalid information is avoided.

We term algorithm resulting from the above modification Prior-Linearization (PL) fixed-lag smoothing. It is important to note that, as illustrated in Fig. 1, the number of variables for which the marginalization process creates prior information is typically small. As a result, typically only a small number of states will be affected by the change of linearization point, and therefore any loss of linearization accuracy (due to the use of older estimates for computing the Jacobians in (24)) is small. As indicated by the results presented in the next section, the effect of this loss of linearization accuracy is not significant, while avoiding the creation of fictitious information leads to significantly improved estimation precision.

IV. Results

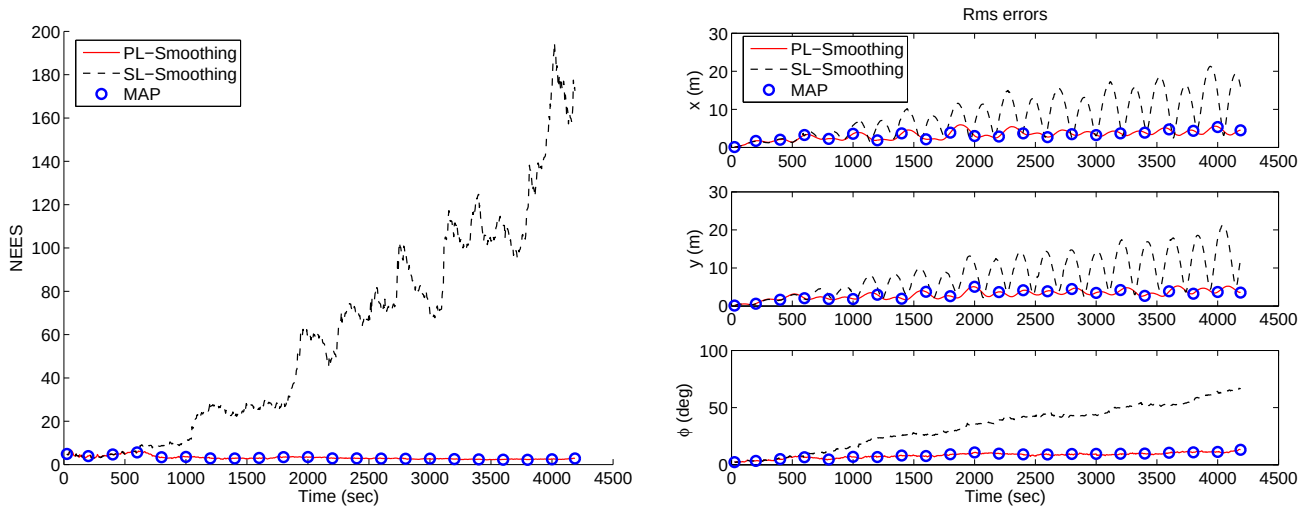
In this section, we present simulation and real-world experimental results that demonstrate the properties of the proposed PL-smoothing algorithm.

A. Simulation results: 2D localization

In our simulation setup, we consider a robot that moves on a plane along a circular trajectory of total length equal to 1200 m. The robot tracks its pose using odometry and bearing measurements to landmarks that lie within its sensing range of 4 m. This scenario could arise, for example, in the case of a robot that moves inside corridors and tracks its position using camera observations of vertical edges on the walls. On average, approximately 15 landmarks are visible at any time, and measurements occur at a rate of 1 Hz in the simulation setup. Landmarks can be tracked for a maximum of 20 consecutive time steps, and therefore in the fixed-lag smoother we choose to maintain a sliding window which contains 25 robot poses and the landmarks seen in these poses. In Fig. 2 the results of the PL-smoothing algorithm are presented, and compared with those obtained by (i) the fixed-lag smoothing algorithm that utilizes the standard linearization approach (termed SL-smoother in the following), and (ii) the full-state MAP estimator.

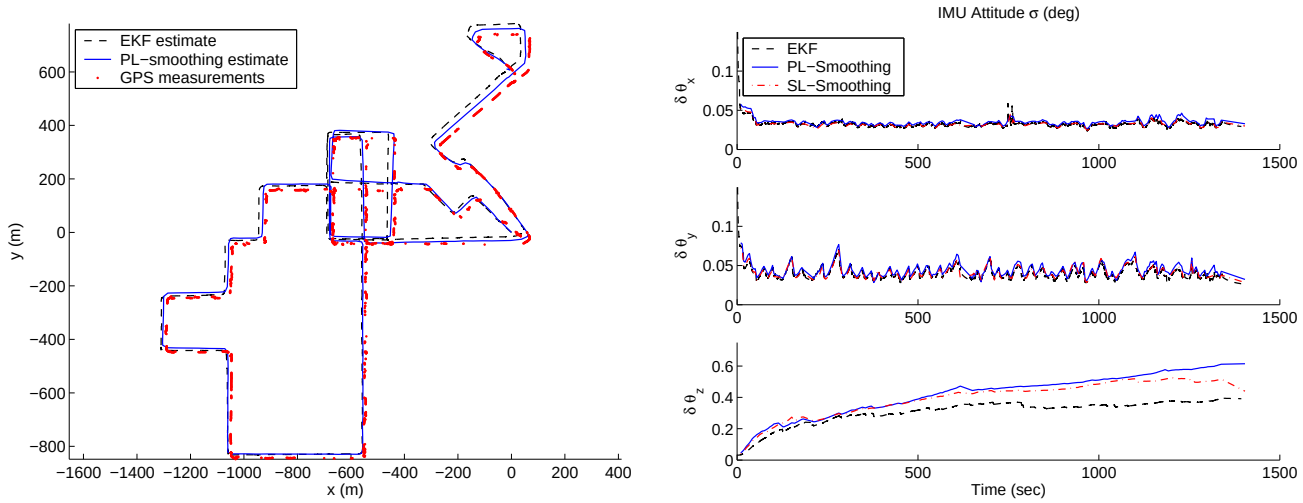
Specifically, Fig. 2(a) shows the average normalized estimation error squared (NEES) for the latest robot pose, averaged over 50 Monte-Carlo runs, while Fig. 2(b) shows the RMS localization errors for each of the three robot states $[x, y, \phi]$. In these plots we observe that the PL-smoothing method significantly outperforms the SL-smoothing approach, both in terms of consistency (i.e., NEES) and in terms of accuracy (i.e., RMS errors). Most importantly, we see that the performance of the PL-smoother is *almost indistinguishable* from that of the full-MAP estimator, which at any time-step carries out estimation using the entire history of states, and all measurements.

The average robot-pose NEES over all Monte-Carlo runs and all time steps equals 3.19 for the full-MAP, and 3.22 for the PL-smoother. (Since in this case the robot state is of dimension 3, the "ideal" NEES value for a consistent estimator equals 3). On the other hand, the average RMS



(a) (b)

Fig. 2. Simulation results for 2D localization using odometry and bearing measurements to features. (a) The average value of the robot-pose NEES over time. (b) The RMS errors for the robot pose over time. In both cases, averaging occurs over all the Monte-Carlo trials. In these plots, the red solid lines correspond to the PL-smoothing algorithm, the black dashed lines to the standard-linearization smoother, while the circles to the full MAP estimator.



(a) (b)

Fig. 3. Real-world results for 3D localization using inertial measurements and a monocular camera. (a) The trajectory estimates vs. GPS ground truth. (b) The reported standard deviation for the 3 axes of rotation.

errors for both estimators are *identical* to three significant digits, equal to 1.38 m for position and 3.61° for orientation. This performance is remarkable, given the fact that the PL-smoother has a computational cost orders of magnitude smaller than that of the full-MAP estimator. Moreover, it becomes clear that the choice of linearization points has a profound effect on both the consistency and the accuracy of the estimates (for comparison, the average NEES for the SL-smoother equals 44.5, while the average position and orientation RMS errors are 1.82 m and 5.58°). We thus see that in the simulation setup shown here, the proposed PL-smoothing is capable of attaining accuracy close to that of the "golden standard" full-state MAP estimator, at a computational cost constant over time and orders of magnitude smaller than that of the full-state MAP.

B. Real-world experiment: 3D localization

To validate the performance of the proposed algorithm in a real-world setting, we tested it on the data collected by a vehicle moving on city streets. The experimental setup consisted of a camera registering images with resolution 640×480 pixels, and an ISIS IMU, providing measurements of rotational velocity and linear acceleration at 100Hz. In this experiment the vehicle drove for about 23 minutes, covering a distance of approximately 8.2 km. Images were processed at a rate of 7.5 Hz, and an average of about 800 features were detected in each image. Features were extracted using the Harris corner detector [20], and matched using normalized cross-correlation.

During the experiment all data were stored on a computer, and processing was carried out off-line, enabling us to test the performance of several methods. Specifically, we compare the performance of the PL-smoother, the SL-smoother, and an EKF-based fixed-lag smoothing method [6]. All three estimators process exactly the same data, and produce estimates of the IMU's 3D pose and velocity, as well as of the IMU's biases. Due to the duration of the dataset, and the number of detected features (approximately 3 million in total), it was impossible to run a full-state MAP estimator on this dataset.

In Fig. 3(a) the trajectory estimates of the PL-smoother and the EKF-smoother are shown in the solid and dashed lines, respectively. Additionally, the dots represent the GPS measurements, which were available intermittently to provide ground truth (GPS was not processed in the estimator). Unfortunately, in this experiment the timestamps of the GPS ground truth were not precise, and therefore it is impossible to compute the exact value of the error for each time instant. However, by inspection of the trajectory estimates, we can deduce that the position errors of the EKF-smoother at the end of the trajectory are approximately double those of the PL-smoother, and are equal to about 0.4% of the traveled distance. The estimates of the SL-smoother are very close to those of the PL-smoother, and they are not shown to preserve the clarity of the figure.

In addition to the estimation errors, it is interesting to examine the accuracy reported by the three algorithms. To this end, in Fig. 3(b) we plot the time evolution of the reported standard deviation for the orientation estimates. The three subplots correspond to the rotation errors about the x, y, and z axes, respectively. We observe that, while the reported accuracies for the rotation about the x and y axes (roll and pitch) are very similar among estimators, those for the rotation about z (yaw) differ significantly. On one hand, the yaw uncertainty reported by the EKF remains almost constant towards the end of the trajectory, and sharply drops for the SL-smoother. On the other hand, the PL-smoother reports that the yaw uncertainty continuously increases. Given that the yaw is unobservable in this experiment, we clearly see that the PL-smoother provides a better representation of the actual uncertainty of the state estimates.

V. Conclusions

In this paper, we presented an algorithm for tracking the motion of a robot using proprioceptive and exteroceptive measurements. The method is based on a fixed-lag smoothing approximation to the full-MAP estimator. In order to attain bounded computational cost over time, the proposed algorithm employs marginalization of older states, so as to maintain a sliding window of active states with approximately constant size. Through an analysis of the marginalization equations, we have proven that if the standard approach to linearization is used (i.e., if the latest estimates of the states are used for computing Jacobians), erroneous information is introduced in the estimator, resulting in inconsistency. To address this problem we have proposed a modified linearization scheme, termed PL-fixed lag smoothing, which prevents the infusion of artificial information, and improves estimation performance. The proposed algorithm was tested in both simulation and real-world experiments, and its accuracy was shown to be superior to that of alternative methods.

Acknowledgments

This work was supported by the University of California, Riverside (BCOE, Regents' Faculty Fellowship).

References

[1] S. I. Roumeliotis, A. E. Johnson, and J. F. Montgomery, "Augmenting inertial navigation with image-based motion estimation," in *Proc. IEEE Intl. Conf. on Robotics and Automation*, Washington D.C, May 2002, pp. 4326–33.

[2] D. D. Diel, "Stochastic constraints for vision-aided inertial navigation," Master's thesis, Massachusetts Institute of Technology, Jan. 2005.

[3] D. S. Bayard and P. B. Brugarolas, "On-board vision-based spacecraft estimation algorithm for small body exploration," *IEEE Trans. on Aerospace and Electonic Systems*, vol. 44, no. 1, pp. 443–460, 2008.

[4] L. H. Matthies, "Dynamic stereo vision," Ph.D. dissertation, School of Computer Science, Carnegie Mellon University, 1989.

[5] A. Howard, "Real-time stereo visual odometry for autonomous ground vehicles," in *Proc. Intl. Conf. on Intelligent Robots and Systems*, Nice, France, Sept. 22-26 2008, pp. 3946–3952.

[6] A. I. Mourikis and S. I. Roumeliotis, "A multi-state constraint Kalman filter for vision-aided inertial navigation," in *Proc. IEEE Intl. Conf. on Robotics and Automation*, Rome, Italy, Apr. 2007, pp. 3565–3572.

[7] P. S. Maybeck, *Stochastic Models, Estimation and Control*, ser. Mathematics in Science and Engineering. London: Academic Press, 1982, vol. 141-2.

[8] D. Nister, O. Naroditsky, and J. Bergen, "Visual odometry for ground vehicle applications," *Journal of Field Robotics*, vol. 23, no. 1, pp. 3–20, Jan. 2006.

[9] C. Engels, H. Stewenius, and D. Nister, "Bundle adjustment rules," in *Proc. Photogrammetric Computer Vision Conf.*, Bonn, Germany, Sep. 20-22 2006, pp. 266–271.

[10] K. Konolige, M. Agrawal, and J. Sola, "Large-scale visual odometry for rough terrain," in *Proc. Intl. Symposium on Research in Robotics*, Hiroshima, Japan, Nov. 26-29 2007.

[11] P. McLauchlan, "The variable state dimension filter applied to surface-based structure from motion," School of Electrical Engineering, Information Technology and Mathematics, University of Surrey, UK, Tech. Rep. VSSP-TR-4/99, 1999.

[12] E. Mouragnon, M. Lhuillier, M. Dhome, F. Dekeyser, and P. Sayd, "Real time localization and 3D reconstruction," in *Proc. IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, June 17-22 2006, pp. 363– 370.

[13] B. Triggs, P. McLauchlan, R. Hartley, and Fitzgibbon, "Bundle adjustment – a modern synthesis," in *Vision Algorithms: Theory and Practice*. Springer Verlag, 2000, pp. 298–375.

[14] G. Sibley, G. S. Sukhatme, and L. Matthies, "Constant time sliding window filter SLAM as a basis for metric visual perception," in *Workshop: From features to actions - Unifying perspectives in computational and robot vision, held at ICRA 2007*, April 2007.

[15] A. Ranganathan, M. Kaess, and F. Dellaert, "Fast 3D pose estimation with out-of-sequence measurements," in *Proc. IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems*, San Diego, CA, Oct. 29 - Nov. 2 2007, pp. 2486–2493.

[16] Y. Bar-Shalom and X. Li, *Estimation and Tracking: Principles, Techniques, and Software*. Boston: Artech House, 1993.

[17] G. Huang, A. I. Mourikis, and S. I. Roumeliotis, "Observability-based rules for designing consistent EKF SLAM estimators," *International Journal of Robotics Research*, vol. 29, no. 5, pp. 502–528, April 2010.

[18] F. Dellaert and M. Kaess, "Square root SAM: Simultaneous localization and mapping via square root information smoothing," *Intl. Journal of Robotics Research*, vol. 25, no. 12, pp. 1181–1203, Dec. 2006.

[19] T.-C. Dong-Si and A. I. Mourikis, "Motion tracking with fixed-lag smoothing: Algorithm and consistency analysis," Dept. of Electrical Engineering, University of California, Riverside, Tech. Rep., 2010, http://www.ee.ucr.edu/∼mourikis/tech_reports/fixed_lag.pdf.

[20] C. Harris and M. Stephens, "A combined corner and edge detector," in *Proc. 4th Alvey Vision Conf.*, Manchester, UK, Aug. 31 - Sep. 2 1988, pp. 147–151.