An Observability-Constrained Sliding Window Filter for SLAM

Guoquan P. Huang, and Stergios I. Roumeliotis

Multiple Autonomous
Robotic Systems Labroratory

Technical Report
Number -2011-0001
February 2011

Dept. of Computer Science & Engineering
University of Minnesota
4-192 EE/CS Building
200 Union St. S.E.
Minneapolis, MN 55455
Tel: (612) 625-2217
Fax: (612) 625-0572
URL: http://www.cs.umn.edu/~ghuang

An Observability-Constrained Sliding Window Filter for SLAM

Guoquan P. Huang, and Stergios I. Roumeliotis

Multiple Autonomous Robotic Systems Laboratory, TR-2011-0001

February 2011

Abstract

A sliding window filter (SWF) is an appealing smoothing algorithm for nonlinear estimation problems such as simultaneous localization and mapping (SLAM), since it is resource-adaptive by controlling the size of the sliding window, and can better address the nonlinearity of the problem by relinearizing available measurements. However, due to the marginalization employed to discard old states from the sliding window, the standard SWF has different parameter observability properties from the optimal batch maximum-a-posterior (MAP) estimator, i.e., the nullspace of the Fisher information matrix (or Hessian) has lower dimension than that of the batch MAP estimator. This implies that the standard SWF acquires spurious information from the available measurements, which can lead to inconsistency. To avoid influx of such nonexistent information, we propose an observability-constrained (OC)-SWF, in which the linearization points for computing Hessian are selected to ensure the correct dimension of the nullspace of the Hessian while minimizing the linearization errors.

1 Introduction and Related Work

Among the existing approaches for robot localization, the extended Kalman filter (EKF) is one of the most popular methods. This is primarily due to its ease of implementation and relatively low processing requirements. However, the EKF, as well as any linearization-based filtering approach, may suffer from the accumulation of linearization errors. This is because once linearization points are selected at a given time step for computing the filter Jacobians, they cannot be updated at later times, when more measurements become available for improving them. In contrast, a batch maximum a posteriori (MAP) estimator [1] can improve the estimation accuracy by computing consistent state estimates for all time steps based on all available measurements. Under a Gaussian prior and measurement noise assumption (which is common in practice), finding the MAP estimates requires solving a nonlinear least-squares problem (see Section 2), whose counterpart in computer vision is known as bundle adjustment [2]. A variety of iterative algorithms have been employed for this problem. For example, the square-root smoothing and mapping (SAM) method [3] solves the SLAM problem efficiently by using variable reordering, a well-known technique for sparse linear systems. However, since the size of the state vector in the batch-MAP estimator increases continuously over time, the processing and memory requirements become too high for real-time operation in large-scale problems (e.g., a robot exploring a large environment with millions of landmarks).

To overcome this limitation, a sliding window filter (SWF) [4] (also called a fixed-lag smoother (FLS) [5–7]) can be used to estimate the states over a sliding time window at a fixed computational cost. The SWF concurrently processes all the measurement constraints between states in the window, and better addresses the nonlinearity of the problem by iteratively relinearizing the process and measurement equations. This approach is resource-adaptive: depending on the available computational resources, it can scale from the iterated EKF solution if only a single time step is maintained, to the optimal batch-MAP solution if the sliding window spans the entire time horizon.

The key characteristic of the SWF is the marginalization of old states from the sliding window, a process that appropriately models the uncertainty of these states [2, 4, 6]. However, due to marginalization, *different*

estimates of the *same states* are used as *linearization points* in computing the Hessian matrix during estimation (see [7]). This results in different parameter observability properties [8] as compared to the batch-MAP estimator. Specifically, the Hessian (Fisher information matrix) of the standard SWF has a nullspace of lower dimension than that of the batch-MAP estimator. This implies that the estimator erroneously *believes* it has information along more directions of the state space than those contained in the measurements. This leads to inconsistent estimates, i.e., estimates whose accuracy is worse than the one reported by the estimator. This inconsistency is a serious problem, since when an estimator is inconsistent, the accuracy of the produced estimates is unknown, which in turn makes the estimator unreliable [8].

In order to improve the consistency and accuracy of the SWF, in this paper we propose an observability-constrained (OC)-SWF as a general smoothing framework. In particular, we postulate that by ensuring the Hessian matrix has a nullspace of appropriate dimension, we can avoid the influx of spurious information in the unobservable directions of the parameter (state) space, thus improving the consistency of the estimates. Based on this, we develop an OC-SWF which extends the observability-based methodology for designing consistent EKFs [9]. The key idea behind our approach is to select the linearization points for computing the Jacobians, and hence the Hessian, so as to ensure that its nullspace dimension does not arbitrarily decrease.

It should be pointed out that a prior-linearization (PL)-SWF for motion estimation was proposed in [7]. In particular, the PL-SWF computes the Hessian using the prior, instead of the current, estimates, for the states connected via measurements to marginalized states. This ensures the same estimates for the same states are used, and the appropriate dimension of the Hessian's nullspace is preserved. However, if the prior estimates are inaccurate, the linearization errors will be large and may degrade the estimator's performance. In contrast, the proposed OC-SWF selects *optimal* linearization points for computing the Hessian, in the sense that they not only ensure the correct dimension for the nullspace of the Hessian, but also minimize the linearization errors. We stress that apart from the SLAM problem treated in this paper, the proposed OC-SWF is applicable to a large class of nonlinear estimation problems in robotics and computer vision, such as visual odometry [10] and vision-aided inertial navigation [11].

2 Problem Formulation

Consider the batch least squares formulation of SLAM [3] where we aim at estimating the entire robot trajectory up to current time step and all the landmarks observed so far. So the state vector is given by:

$$\mathbf{x}_{0:k} = \begin{bmatrix} \mathbf{x}_{R_0}^T & \mathbf{x}_{R_1}^T & \cdots & \mathbf{x}_{R_k}^T & \mathbf{p}_{L_1}^T & \cdots & \mathbf{p}_{L_M}^T \end{bmatrix}^T \tag{1}$$

where $\mathbf{x}_{R_k} = [\mathbf{p}_{R_k}^T \quad \phi_{R_k}]^T$ denotes the robot pose, and $\mathbf{p}_{L_i}$ is the i-th landmark position. In the following, we start by presenting the *generalized* robot motion model and sensor measurement model that will be used throughout the paper.

Consider a robot equipped with an odometry sensor moving on a plane. The odometry serves as the control input to propagate the robot pose, according to the following motion model:

$$\mathbf{p}_{R_k} = \mathbf{p}_{R_{k-1}} + \mathbf{C}(\phi_{R_{k-1}}) \, {}^{R_{k-1}}\mathbf{p}_{R_k} \tag{2}$$

$$\phi_{R_k} = \phi_{R_{k-1}} + {}^{R_{k-1}}\phi_{R_k} \tag{3}$$

where $\mathbf{C}(\cdot)$ denotes the 2×2 rotation matrix, and $\mathbf{u}_{k-1} = {}^{R_{k-1}}\mathbf{x}_{R_k} = [{}^{R_{k-1}}\mathbf{p}_{R_k}^T \quad {}^{R_{k-1}}\phi_{R_k}]^T$ is the true odometry (control input), i.e., the robot's motion between time-steps $k-1$ and k, expressed with respect to the robot's frame at time-step $k-1$, $\{R_{k-1}\}$. The corresponding odometry measurement, $\mathbf{u}_{m_{k-1}}$, is assumed to be corrupted by zero-mean, white Gaussian noise, $\mathbf{w}_{k-1} = \mathbf{u}_{k-1} - \mathbf{u}_{m_{k-1}}$, with covariance $\mathbf{Q}_{k-1}$. This motion model is described by the following generic nonlinear function:

$$\mathbf{g}(\mathbf{x}_{0:k}, \mathbf{u}_{k-1}) = \mathbf{x}_{R_k} - \mathbf{f}(\mathbf{x}_{R_{k-1}}, \mathbf{u}_{m_{k-1}} + \mathbf{w}_{k-1}) = \mathbf{0} \tag{4}$$

To employ a batch-MAP estimator, it is necessary to linearize (4) and compute the Jacobians with respect

to the state vector (1) and the noise, respectively, i.e.,

$$\mathbf{\Phi}_{k-1} \triangleq \frac{\partial \mathbf{g}}{\partial \mathbf{x}_{0:k}}\Big|_{\{\mathbf{x}^\star_{0:k},\mathbf{0}\}} = \begin{bmatrix}\mathbf{0}_{3\times 3} & \cdots & \mathbf{\Phi}_{R_{k-1}} & \mathbf{I}_3 & \mathbf{0}_{3\times 2} & \cdots & \mathbf{0}_{3\times 2}\end{bmatrix} \tag{5}$$

$$\mathbf{G}_{k-1} \triangleq \frac{\partial \mathbf{g}}{\partial \mathbf{w}_{k-1}}\Big|_{\{\mathbf{x}^\star_{0:k},\mathbf{0}\}} = \begin{bmatrix}\mathbf{C}(\phi^\star_{R_{k-1}}) & \mathbf{0}_{2\times 1} \\ \mathbf{0}_{1\times 2} & 1\end{bmatrix} \tag{6}$$

with

$$\mathbf{\Phi}_{R_{k-1}} = -\begin{bmatrix}\mathbf{I}_2 & \mathbf{J}\left(\mathbf{p}^\star_{R_k} - \mathbf{p}^\star_{R_{k-1}}\right) \\ \mathbf{0}_{1\times 2} & 1\end{bmatrix} \tag{7}$$

where $\mathbf{x}^\star_{0:k}$ denotes the linearization point for the state (1), while a zero vector is used as the linearization point for the noise, and $\mathbf{J} = \begin{bmatrix}0 & -1 \\ 1 & 0\end{bmatrix}$. Clearly, the values of the Jacobian matrices depend on the choice of linearization points, which is the key fact our approach relies on. Note also that the form of the motion model presented above is general, and holds for any robot kinematic model (e.g., unicycle, bicycle, or Ackerman model) [9].

During SLAM, the robot-to-landmark measurements are a function of the relative position of the observed landmark with respect to the robot:

$$\mathbf{z}_{ij} = \mathbf{h}_{ij}(\mathbf{x}_{0:k}) + \mathbf{v}_{ij} = \mathbf{h}\left({}^{R_j}\mathbf{p}_{L_i}\right) + \mathbf{v}_{ij} \tag{8}$$

where ${}^{R_j}\mathbf{p}_{L_i} = \mathbf{C}^T(\phi_{R_j})(\mathbf{p}_{L_i} - \mathbf{p}_{R_j})$ is the position of the i-th landmark with respect to the robot at time-step j, and $\mathbf{v}_{ij}$ is zero-mean Gaussian measurement noise with covariance $\mathbf{R}_{ij}$. In this work, we allow $\mathbf{h}(\cdot)$ to be *any* measurement function (e.g., a direct measurement of relative position, a pair of range and bearing measurements, bearing-only measurements, etc.). In general, the measurement function is nonlinear, and its Jacobian matrix is given by:

$$\mathbf{H}_{ij} \triangleq \frac{\partial \mathbf{h}_{ij}}{\partial \mathbf{x}_{0:k}}\Big|_{\{\mathbf{x}^\star_{0:k},\mathbf{0}\}} = \begin{bmatrix}\mathbf{0} & \cdots & \mathbf{H}_{R_{ij}} & \mathbf{0} & \cdots & \mathbf{H}_{L_{ij}} & \mathbf{0} & \cdots & \mathbf{0}\end{bmatrix} \tag{9}$$

with

$$\mathbf{H}_{R_{ij}} = (\nabla \mathbf{h}_{ij})\mathbf{C}^T(\phi^\star_{R_j})\begin{bmatrix}-\mathbf{I}_2 & -\mathbf{J}(\mathbf{p}^\star_{L_i} - \mathbf{p}^\star_{R_j})\end{bmatrix} \tag{10}$$

$$\mathbf{H}_{L_{ij}} = (\nabla \mathbf{h}_{ij})\mathbf{C}^T(\phi^\star_{R_j}) \tag{11}$$

where $\mathbf{H}_{R_{ij}}$ and $\mathbf{H}_{L_{ij}}$ are the Jacobians with respect to the robot pose at time-step j and the i-th landmark position, respectively, and $\nabla \mathbf{h}_{ij}$ denotes the Jacobian of $\mathbf{h}_{ij}$ with respect to the robot-relative landmark position, ${}^{R_j}\mathbf{p}_{L_i}$, evaluated at the linearization point, $\mathbf{x}^\star_{0:k}$.

2.1 Batch MAP Estimator

The batch-MAP estimator utilizes all the available information to estimate the state vector (1). The information used includes: (i) the prior information about the initial state, described by a Gaussian pdf with mean $\hat{\mathbf{x}}_{0|0}$ and covariance $\mathbf{P}_{0|0}$, (ii) the motion information (4), and (iii) the sensor measurements (8). In particular, the batch-MAP estimator seeks to determine the estimate $\hat{\mathbf{x}}_{0:k|k}$ that maximizes the posterior pdf:

$$p(\mathbf{x}_{0:k}|\mathcal{Z}_{0:k}) \propto p(\mathbf{x}_{R_0})\prod_{\kappa=1}^{k} p(\mathbf{x}_{R_\kappa}|\mathbf{x}_{R_{\kappa-1}}) \prod_{\mathbf{z}_{ij}\in\mathcal{Z}_{0:k}} p(\mathbf{z}_{ij}|\mathbf{x}_{R_j}, \mathbf{p}_{L_i}) \tag{12}$$

where $\mathcal{Z}_{0:k}$ denotes all the available measurements in the time interval $[0,k]$. For Gaussian and independent state and measurement noise (see (4), and (8), respectively), this pdf (12) can be written as:

$$\begin{aligned} & p(\mathbf{x}_{0:k}|\mathbf{z}_{0:k}) \propto \\ & \frac{1}{\sqrt{(2\pi)^n|\mathbf{P}_{0|0}|}}\exp\left(-\frac{1}{2}||\mathbf{x}_{R_0}-\hat{\mathbf{x}}_{0|0}||^2_{\mathbf{P}_{0|0}}\right)\times \\ & \prod_{\kappa=1}^{k}\frac{1}{\sqrt{(2\pi)^3|\mathbf{Q}'_{\kappa-1}|}}\exp\left(-\frac{1}{2}||\mathbf{x}_\kappa-\mathbf{f}(\mathbf{x}_{\kappa-1},\mathbf{u}_{m_{\kappa-1}})||^2_{\mathbf{Q}'_{\kappa-1}}\right)\times \\ & \prod_{\mathbf{z}_{ij}\in\mathcal{Z}_{0:k}}\frac{1}{\sqrt{(2\pi)^m|\mathbf{R}_{ij}|}}\exp\left(-\frac{1}{2}||\mathbf{z}_{ij}-\mathbf{h}_{ij}(\mathbf{x}_{0:k})||^2_{\mathbf{R}_{ij}}\right) \end{aligned} \tag{13}$$

where n is the dimension of the prior state $\mathbf{x}_{R_0}$, and $m=1$ or 2 is the dimension of the measurement $\mathbf{z}_{ij}$. In the above expression, we have also employed the notations, $||\mathbf{a}||^2_{\mathbf{M}} \triangleq \mathbf{a}^T\mathbf{M}^{-1}\mathbf{a}$ and $\mathbf{Q}'_k \triangleq \mathbf{G}_k\mathbf{Q}_k\mathbf{G}_k^T$ (see (4)). Thus, the maximization of (13) is equivalent to the minimization of the following cost function:

$$c(\mathbf{x}_{0:k}) = \frac{1}{2}||\mathbf{x}_{R_0}-\hat{\mathbf{x}}_{0|0}||^2_{\mathbf{P}_{0|0}} + \sum_{\kappa=1}^{k}\frac{1}{2}||\mathbf{x}_{R_\kappa}-\mathbf{f}(\mathbf{x}_{R_{\kappa-1}},\mathbf{u}_{m_{\kappa-1}})||^2_{\mathbf{Q}'_{\kappa-1}} + \sum_{\mathbf{z}_{ij}\in\mathcal{Z}_{0:k}}\frac{1}{2}||\mathbf{z}_{ij}-\mathbf{h}_{ij}(\mathbf{x}_{0:k})||^2_{\mathbf{R}_{ij}} \tag{14}$$

$c(\mathbf{x}_{0:k})$ is a nonlinear function, and a standard approach to determine its minimum is to employ Newton-Raphson iterative minimization [2]. Specifically, at the ℓ-th iteration of this method, a correction, $\delta\mathbf{x}^{(\ell)}_{0:k}$, to the current estimate, $\hat{\mathbf{x}}^{(\ell)}_{0:k|k}$, is computed by minimizing the second-order Taylor-series approximation of the cost function which is given by:

$$c(\hat{\mathbf{x}}^{(\ell)}_{0:k|k}+\delta\mathbf{x}^{(\ell)}_{0:k}) \simeq c(\hat{\mathbf{x}}^{(\ell)}_{0:k|k}) + \mathbf{b}_b^{(\ell)^T}\delta\mathbf{x}^{(\ell)}_{0:k} + \frac{1}{2}\delta\mathbf{x}^{(\ell)^T}_{0:k}\mathbf{A}_b^{(\ell)}\delta\mathbf{x}^{(\ell)}_{0:k} \tag{15}$$

where

$$\mathbf{b}_b^{(\ell)} \triangleq \nabla_{\mathbf{x}_{0:k}}c(\cdot)\Big|_{\{\mathbf{x}^\star_{0:k}=\hat{\mathbf{x}}^{(\ell)}_{0:k|k}\}} \tag{16}$$

$$\mathbf{A}_b^{(\ell)} \triangleq \nabla^2_{\mathbf{x}_{0:k}}c(\cdot)\Big|_{\{\mathbf{x}^\star_{0:k}=\hat{\mathbf{x}}^{(\ell)}_{0:k|k}\}} \tag{17}$$

are the gradient and Hessian of $c(\cdot)$ with respect to $\mathbf{x}_{0:k}$, evaluated at the current state estimate $\hat{\mathbf{x}}^{(\ell)}_{0:k|k}$.

We now examine the structure of the Jacobian and Hessian matrices which will be used in the ensuing analysis. Specifically, at the ℓ-th iteration, $\mathbf{b}_b^{(\ell)}$ is (see (5) and (9)):

$$\mathbf{b}_b^{(\ell)} = \mathbf{\Pi}^T\mathbf{P}^{-1}_{0|0}\left(\hat{\mathbf{x}}^{(\ell)}_{R_{0|k}}-\hat{\mathbf{x}}_{0|0}\right) + \sum_{\kappa=1}^{k}\mathbf{\Phi}^{(\ell)^T}_{\kappa-1}\mathbf{Q}'^{-1}_{\kappa-1}\left(\hat{\mathbf{x}}^{(\ell)}_{R_{\kappa|k}}-\mathbf{f}(\hat{\mathbf{x}}^{(\ell)}_{R_{\kappa-1|k}},\mathbf{u}_{m_{\kappa-1}})\right) - \sum_{\mathbf{z}_{ij}\in\mathcal{Z}_{0:k}}\mathbf{H}_{ij}^{(\ell)^T}\mathbf{R}^{-1}_{ij}\left(\mathbf{z}_{ij}-\mathbf{h}_{ij}(\hat{\mathbf{x}}^{(\ell)}_{0:k|k})\right) \tag{18}$$

where $\mathbf{\Pi} = \begin{bmatrix}\mathbf{I}_n & \mathbf{0} & \cdots & \mathbf{0}\end{bmatrix}$ and again $n=\dim(\mathbf{x}_{R_0})$. On the other hand, the Hessian matrix, $\mathbf{A}_b^{(\ell)}$, is approximated in the Gauss-Newton method by (see (5) and (9)):

$$\mathbf{A}_b^{(\ell)} = \mathbf{\Pi}^T\mathbf{P}^{-1}_{0|0}\mathbf{\Pi} + \sum_{\kappa=1}^{k}\mathbf{\Phi}^{(\ell)^T}_{\kappa-1}\mathbf{Q}'^{-1}_{\kappa-1}\mathbf{\Phi}^{(\ell)}_{\kappa-1} + \sum_{\mathbf{z}_{ij}\in\mathcal{Z}_{0:k}}\mathbf{H}_{ij}^{(\ell)^T}\mathbf{R}^{-1}_{ij}\mathbf{H}^{(\ell)}_{ij} \tag{19}$$

which is a good approximation for small-residual problems [2]. Due to the sparse structure of the matrices $\mathbf{H}^{(\ell)}_{ij}$ and $\mathbf{\Phi}^{(\ell)}_\kappa$ (see (5) and (9)), the matrix $\mathbf{A}_b^{(\ell)}$ is also sparse, which can be exploited to speed-up the solution of the linear system in (20) [2]. The value $\delta\mathbf{x}^{(\ell)}_{0:k}$ that minimizes (15) is found by solving the following linear system:

$$\mathbf{A}_b^{(\ell)}\delta\mathbf{x}^{(\ell)}_{0:k} = -\mathbf{b}_b^{(\ell)} \tag{20}$$

Once $\delta\mathbf{x}_{0:k}^{(\ell)}$ is found, the new state estimate is computed as:

$$\hat{\mathbf{x}}_{0:k|k}^{(\ell+1)} = \hat{\mathbf{x}}_{0:k|k}^{(\ell)} + \delta\mathbf{x}_{0:k}^{(\ell)} \tag{21}$$

Given an initial estimate $\hat{\mathbf{x}}_{0:k|k}^{(0)}$ that resides within the attraction basin of the global optimum, this iterative algorithm will compute the global minimum (i.e., MAP estimate) for the entire state given all measurements up to time-step k.

2.2 Standard Sliding-Window Filter

As the robot continuously moves and observes new landmarks, the size of the state vector $\mathbf{x}_{0:k}$ in the batch MAP estimator constantly increases (linearly in time). Clearly, this is not always suitable for real-time operations. To reduce the computational cost of the batch MAP estimator, the SWF employs a marginalization process to optimally discard old, matured states and thus only maintains a constant-size window of states. In what follows, we present the detailed derivations of the standard SWF that forms the basis for our proposed approach. Note that similar derivations can also be found int [7].

In particular, suppose that during the time interval $[0, k]$, all available measurements are collected, and a batch MAP estimation is carried out at time-step k. Some notations are first explained: The old states:

$$\mathbf{x}_\mathbf{M} \triangleq \begin{bmatrix} \mathbf{x}_{R_{0:m}}^T & \mathbf{p}_{L_{M_1}}^T & \cdots & \mathbf{p}_{L_{M_m}}^T \end{bmatrix}^T$$

that are the states to be marginalized out. Note that it is not necessary to sequentially marginalize out the old robot poses $\mathbf{x}_{R_{0:m}}$, instead we can choose and discard the most matured ones. The remaining states:

$$\mathbf{x}_\mathbf{R} \triangleq \begin{bmatrix} \mathbf{x}_{R_{m+1:k}}^T & \mathbf{p}_{L_{R_1}}^T & \cdots & \mathbf{p}_{L_{R_r}}^T \end{bmatrix}^T$$

that remain active in the sliding window. Upon marginalization, all the states in $\mathbf{x}_\mathbf{M}$, as well as all the measurements that involve these states (denoted by $\mathcal{Z}_M$) are discarded. Then, as the robot keeps moving and the sensor continuously collects new measurements in the time interval $[k, k']$, the state vector is augmented by the new robot states and the new observed landmarks,

$$\mathbf{x}_\mathbf{N} \triangleq \begin{bmatrix} \mathbf{x}_{R_{k+1:k'}}^T & \mathbf{p}_{L_{N_1}}^T & \cdots & \mathbf{p}_{L_{N_n}}^T \end{bmatrix}^T$$

Thus, at time-step k', the sliding window contains the states $\mathbf{x}_\mathbf{R}$ and $\mathbf{x}_\mathbf{N}$. In particular, the batch MAP estimate (i.e., without marginalization) is computed by minimizing a cost function similar to (14):

$$c(\mathbf{x}_{0:k'}) = c(\mathbf{x}_\mathbf{M}, \mathbf{x}_\mathbf{R}, \mathbf{x}_\mathbf{N}) = c_m(\mathbf{x}_\mathbf{M}, \mathbf{x}_\mathbf{R}) + c_n(\mathbf{x}_\mathbf{R}, \mathbf{x}_\mathbf{N}) \tag{22}$$

where we have decomposed the cost function into two terms: $c_m(\mathbf{x}_\mathbf{M}, \mathbf{x}_\mathbf{R})$ that contains all quadratic terms that involve states in $\mathbf{x}_\mathbf{M}$ only, as well as terms involving one state in $\mathbf{x}_\mathbf{M}$ and one in $\mathbf{x}_\mathbf{R}$; and $c_n(\mathbf{x}_\mathbf{R}, \mathbf{x}_\mathbf{N})$ that contains all quadratic terms that involve states in $\mathbf{x}_\mathbf{R}$ only, states in $\mathbf{x}_\mathbf{N}$ only, or terms involving one state in $\mathbf{x}_\mathbf{R}$ and one in $\mathbf{x}_\mathbf{N}$. It is important to note that, no quadratic terms jointly involving states in $\mathbf{x}_\mathbf{N}$ and $\mathbf{x}_\mathbf{M}$ exist, since the states marginalized at time-step k do not participate in any measurement after that time.[1] Thus, we have:

$$\min_{\mathbf{x}_\mathbf{M}, \mathbf{x}_\mathbf{R}, \mathbf{x}_\mathbf{N}} c(\mathbf{x}_\mathbf{M}, \mathbf{x}_\mathbf{R}, \mathbf{x}_\mathbf{N}) = \min_{\mathbf{x}_\mathbf{R}, \mathbf{x}_\mathbf{N}} \left(\min_{\mathbf{x}_\mathbf{M}} c(\mathbf{x}_\mathbf{M}, \mathbf{x}_\mathbf{R}, \mathbf{x}_\mathbf{N}) \right) = \min_{\mathbf{x}_\mathbf{R}, \mathbf{x}_\mathbf{N}} \left(c_n(\mathbf{x}_\mathbf{R}, \mathbf{x}_\mathbf{N}) + \min_{\mathbf{x}_\mathbf{M}} c_m(\mathbf{x}_\mathbf{M}, \mathbf{x}_\mathbf{R}) \right) \tag{23}$$

Now, we solve the minimization of c_m (see (14)):

$$c_m(\mathbf{x}_\mathbf{M}, \mathbf{x}_\mathbf{R}) = \frac{1}{2}||\mathbf{x}_0 - \hat{\mathbf{x}}_{0|0}||_{\mathbf{P}_{0|0}}^2 + \sum_{\mathbf{z}_{ij} \in \mathcal{Z}_M} \frac{1}{2}||\mathbf{z}_{ij} - \mathbf{h}_{ij}(\mathbf{x}_{0:k})||_{\mathbf{R}_{ij}}^2 + \sum_{\kappa=1}^{m+1} \frac{1}{2}||\mathbf{x}_\kappa - \mathbf{f}(\mathbf{x}_{\kappa-1}, \mathbf{u}_{m_{\kappa-1}})||_{\mathbf{Q}'_{\kappa-1}}^2 \tag{24}$$

[1] This requires a careful selection of the states to marginalize out, since, otherwise, an approximation will be introduced, e.g., when a new robot pose in $\mathbf{x}_\mathbf{N}$ measures a discarded landmark in $\mathbf{x}_\mathbf{M}$ which occurs often in the case of loop closure.

Clearly, c_m contains the quadratic cost terms due to (i) (if available) the prior, (ii) the robot motion model involving the marginalized states and the states in $\mathbf{x_R}$, and (iii) the measurements that involve the marginalized states. Due to the nonlinearity of the motion and measurement models, the second-order Taylor-series approximation to c_m is employed:

$$c_m \simeq c_m(\hat{\mathbf{x}}_{\mathbf{M}_{k|k}}, \hat{\mathbf{x}}_{\mathbf{R}_{k|k}}) + \mathbf{b}_m^T \begin{bmatrix} \mathbf{x_M} - \hat{\mathbf{x}}_{\mathbf{M}_{k|k}} \\ \mathbf{x_R} - \hat{\mathbf{x}}_{\mathbf{R}_{k|k}} \end{bmatrix} + \frac{1}{2} \begin{bmatrix} \mathbf{x_M} - \hat{\mathbf{x}}_{\mathbf{M}_{k|k}} \\ \mathbf{x_R} - \hat{\mathbf{x}}_{\mathbf{R}_{k|k}} \end{bmatrix}^T \mathbf{A}_m \begin{bmatrix} \mathbf{x_M} - \hat{\mathbf{x}}_{\mathbf{M}_{k|k}} \\ \mathbf{x_R} - \hat{\mathbf{x}}_{\mathbf{R}_{k|k}} \end{bmatrix} \tag{25}$$

where $\mathbf{b}_m$ and $\mathbf{A}_m$ are the Jacobian and Hessian matrices of c_m, evaluated at the MAP estimates of $\mathbf{x_M}$ and $\mathbf{x_R}$ at time-step k (see (18) and (19)), which are the best available estimates since minimization takes place at time-step k. We note that the following block partitioning of the Jacobian and Hessian matrices of c_m will be useful for ensuing derivations:

$$\mathbf{b}_m = \nabla_{\{\mathbf{x}_{\mathbf{M}_k}, \mathbf{x}_{\mathbf{R}_k}\}} c_m(\cdot)\Big|_{\{\mathbf{x}^\star_{\mathbf{M}_k} = \hat{\mathbf{x}}_{\mathbf{M}_{k|k}}, \mathbf{x}^\star_{\mathbf{R}_k} = \hat{\mathbf{x}}_{\mathbf{R}_{k|k}}\}} = \begin{bmatrix} \mathbf{b}_{mm_k} \\ \mathbf{b}_{mr_k} \end{bmatrix} \tag{26}$$

$$\mathbf{A}_m = \nabla^2_{\{\mathbf{x}_{\mathbf{M}_k}, \mathbf{x}_{\mathbf{R}_k}\}} c_m(\cdot)\Big|_{\{\mathbf{x}^\star_{\mathbf{M}_k} = \hat{\mathbf{x}}_{\mathbf{M}_{k|k}}, \mathbf{x}^\star_{\mathbf{R}_k} = \hat{\mathbf{x}}_{\mathbf{R}_{k|k}}\}} = \begin{bmatrix} \mathbf{A}_{mm_k} & \mathbf{A}_{mr_k} \\ \mathbf{A}_{rm_k} & \mathbf{A}_{rr_k} \end{bmatrix} \tag{27}$$

where the dimensions of the blocks correspond to the dimensions of $\mathbf{x_M}$ and $\mathbf{x_R}$, and the subscript k denotes the fact that all quantities in these matrices are evaluated using the state estimates at time-step k. We note that the cost function in (25) is a quadratic function of $\mathbf{x_M}$, and thus the optimal value of $\mathbf{x_M}$ can be attained as:

$$\mathbf{x_M} = \hat{\mathbf{x}}_{\mathbf{M}_{k|k}} - \mathbf{A}^{-1}_{mm_k} \left(\mathbf{b}_{mm_k} + \mathbf{A}_{mr_k} (\mathbf{x_R} - \hat{\mathbf{x}}_{\mathbf{R}_{k|k}}) \right) \tag{28}$$

And substitution in (25) yields the minimum value of c_m:

$$\min_{\mathbf{x_M}} c_m(\mathbf{x_M}, \mathbf{x_R}) \simeq \alpha + \mathbf{b}^T_{p_k} (\mathbf{x_R} - \hat{\mathbf{x}}_{\mathbf{R}_{k|k}}) + \frac{1}{2} (\mathbf{x_R} - \hat{\mathbf{x}}_{\mathbf{R}_{k|k}})^T \mathbf{A}_{p_k} (\mathbf{x_R} - \hat{\mathbf{x}}_{\mathbf{R}_{k|k}}) \tag{29}$$

where α is a constant independent of $\mathbf{x_R}$ and $\mathbf{x_M}$, and

$$\mathbf{b}_{p_k} = \mathbf{b}_{mr_k} - \mathbf{A}_{rm_k} \mathbf{A}^{-1}_{mm_k} \mathbf{b}_{mm_k} \tag{30}$$

$$\mathbf{A}_{p_k} = \mathbf{A}_{rr_k} - \mathbf{A}_{rm_k} \mathbf{A}^{-1}_{mm_k} \mathbf{A}_{mr_k} \tag{31}$$

Thus, with (29) and (23), 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 of the following cost function:

$$\begin{aligned} c'_n(\mathbf{x_R}, \mathbf{x_N}) = \; & \mathbf{b}^T_{p_k} (\mathbf{x_R} - \hat{\mathbf{x}}_{\mathbf{R}_{k|k}}) + \frac{1}{2} (\mathbf{x_R} - \hat{\mathbf{x}}_{\mathbf{R}_{k|k}})^T \mathbf{A}_{p_k} (\mathbf{x_R} - \hat{\mathbf{x}}_{\mathbf{R}_{k|k}}) + \\ & \sum_{\mathbf{z}_{ij} \in \mathcal{Z}_A} \frac{1}{2} ||\mathbf{z}_{ij} - \mathbf{h}(\mathbf{x}_{0:k'})||^2_{\mathbf{R}_{ij}} + \sum_{\kappa=m}^{k'} \frac{1}{2} ||\mathbf{x}_{\kappa+1} - \mathbf{f}(\mathbf{x}_\kappa, \mathbf{u}_{m_\kappa})||^2_{\mathbf{Q}'_\kappa} \end{aligned} \tag{32}$$

where all available measurements up to time-step k' is decomposed into two sets: the active set of measurements whose involved states are all in $\mathbf{x_R}$ and $\mathbf{x_N}$, and the inactive set of measurements at least one of whose states is marginalized, i.e., $\mathcal{Z}_{0:k'} = \mathcal{Z}_A \cup \mathcal{Z}_M$. It is important to note that the above cost function does not depend on $\mathbf{x_M}$. The approximation here lies in the fact that the term c_m has been permanently approximated by the second-order Taylor series approximation of (25). This will introduce small errors in the MAP estimates for $\mathbf{x_R}$ and $\mathbf{x_N}$, but if the marginalized states are old, "mature" ones, with good estimates, the effect of the approximation will be small. On the other hand, the gain from employing this approximation is that the marginalized states $\mathbf{x_M}$, as well as all measurements that directly involve them, can be discarded, thus yielding an algorithm with constant-time and constant-memory requirements. In particular, as proceeding in the full-state batch MAP, the minimization of $c'_n(\mathbf{x_R}, \mathbf{x_N})$ at time-step k' (see (32)), can be carried out by the Gauss-Newton method (see Section 2.1).

Specifically, 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 $\mathbf{A}^{(\ell)}\delta\mathbf{x}_{k'}^{(\ell)} = -\mathbf{b}^{(\ell)}$, where the Jacobian and Hessian matrices are computed by (see (5) and (9)):

$$\mathbf{b}^{(\ell)} = \mathbf{\Pi}_r^T\mathbf{b}_{p_k} + \mathbf{\Pi}_r^T\mathbf{A}_{p_k}\left(\mathbf{x}_{\mathbf{M}}^{(\ell)} - \hat{\mathbf{x}}_{\mathbf{M}_{k|k}}\right) + \sum_{\mathbf{z}_{ij}\in\mathcal{Z}_A}\mathbf{H}_{ij}^{(\ell)^T}\mathbf{R}_{ij}^{-1}\left(\mathbf{z}_{ij} - \mathbf{h}_{ij}(\mathbf{x}_{0:k'})\right) + \sum_{\kappa=m}^{k'}\mathbf{\Phi}_\kappa^{(\ell)^T}\mathbf{Q}_\kappa'^{-1}\left(\mathbf{x}_{\kappa+1}^{(\ell)} - \mathbf{f}(\mathbf{x}_\kappa^{(\ell)}, \mathbf{u}_{m_\kappa})\right) \tag{33}$$

$$\mathbf{A}^{(\ell)} = \mathbf{\Pi}_r^T\mathbf{A}_{p_k}\mathbf{\Pi}_r + \sum_{\mathbf{z}_{ij}\in\mathcal{Z}_A}\mathbf{H}_{ij}^{(\ell)^T}\mathbf{R}_{ij}^{-1}\mathbf{H}_{ij}^{(\ell)} + \sum_{\kappa=m}^{k'}\mathbf{\Phi}_\kappa^{(\ell)^T}\mathbf{Q}_\kappa'^{-1}\mathbf{\Phi}_\kappa^{(\ell)} \tag{34}$$

where $\mathbf{\Pi}_r = \begin{bmatrix}\mathbf{I}_r & \mathbf{0} & \cdots & \mathbf{0}\end{bmatrix}$ and $r = \dim(\mathbf{x_R})$. It is very important to note that in the above expressions, all the involved Jacobians, $\mathbf{\Phi}_\kappa$ and $\mathbf{H}_\kappa$, are evaluated using all the available measurements up to time-step k'. Compared to the structure of these equations with those of (18) and (19), which correspond to the full-state batch MAP estimator, it becomes clear that the term $\mathbf{A}_{p_k}$ expresses the prior information about the states $\mathbf{x_R}$, which arises due to the marginalization of $\mathbf{x_M}$ at time step k. An important difference of this set of equations compared to those of (18) and (19) is the fact that in (33), the prior information is expressed with two terms: in particular, the constant term $\mathbf{b}_{p_k}$ appears in (33), while no such constant term appears in (18) (put differently, the constant term in (18) is zero). This "extra term" in (33) is a consequence of the fact that the prior estimate $\hat{\mathbf{x}}_{\mathbf{R}_{k|k}}$ is not the optimal estimate that would arise using the marginalized measurements. To see why this is the case, we note that the cost function cm in (24) does not have a minimum at $\hat{\mathbf{x}}_{\mathbf{M}_{k|k}}$ and $\hat{\mathbf{x}}_{\mathbf{R}_{k|k}}$, since its Jacobian, $\mathbf{b}_{m_k}$ is generally not zero at $\hat{\mathbf{x}}_{\mathbf{M}_{k|k}}$ and $\hat{\mathbf{x}}_{\mathbf{R}_{k|k}}$. This occurs because the estimates, $\hat{\mathbf{x}}_{\mathbf{M}_{k|k}}$ and $\hat{\mathbf{x}}_{\mathbf{R}_{k|k}}$, have been computed using not only the discarded measurements, which define the cost function c_m, but all the available measurements at time-step k.

2.3 Parameter Observability Analysis

In this section, we perform parameter observability analysis [8] for the batch MAP-based SLAM and the standard SWF-based SLAM, respectively. By comparing their observability properties, we draw conclusions about the estimator consistency. Note that similar analysis has appeared in [7]. The parameter observability indicates the amount of information can be acquired from the available measurements, which is encapsulated by the Fisher information matrix (Hessian). Thus, in the following we will examine the nullspaces of the Hessian matrices of the batch MAP and standard SWF, respectively. Note that we are interested in the information contained in the (proprioceptive and exteroceptive) measurements, without considering the prior (i.e., $\mathbf{P}_{0|0} = \infty \Leftrightarrow \mathbf{P}_{0|0}^{-1} = \mathbf{0}$).

In the batch MAP estimator, the Hessian matrix denoted by $\mathbf{A}_b$, at time-step k' is given by (see (19)):

$$\mathbf{A}_b(k') = \sum_{\mathbf{z}_{ij}\in\mathcal{Z}_M}\mathbf{H}_{ij}^T(k')\mathbf{R}_{ij}^{-1}\mathbf{H}_{ij}(k') + \sum_{\mathbf{z}_{ij}\in\mathcal{Z}_A}\mathbf{H}_{ij}^T(k')\mathbf{R}_{ij}^{-1}\mathbf{H}_{ij}(k') + \sum_{\kappa=1}^{k'}\mathbf{\Phi}_\kappa^T(k')\mathbf{Q}_\kappa'^{-1}\mathbf{\Phi}_\kappa(k') \tag{35}$$

which can be further partitioned as follows:

$$\mathbf{A}_b(k') = \begin{bmatrix}\mathbf{A}_{mm}(k') & \mathbf{A}_{mr}(k') & \mathbf{0}\\ \mathbf{A}_{rm}(k') & \mathbf{A}_{rr}(k') & \mathbf{A}_{rn}(k')\\ \mathbf{0} & \mathbf{A}_{nr}(k') & \mathbf{A}_{nn}(k')\end{bmatrix} \tag{36}$$

Note that all the Jacobians involved in $\mathbf{A}_b(k')$ are computed using the latest state estimates, $\hat{\mathbf{x}}_{0:k'|k'}$. By

inspection, we can obtain one basis of its right nullspace given by

$$\text{null}(\mathbf{A}_b(k')) = \begin{bmatrix} \mathbf{I} & \mathbf{J}\hat{\mathbf{p}}_{R_{0|k'}} \\ \mathbf{0} & 1 \\ \vdots & \vdots \\ \mathbf{I} & \mathbf{J}\hat{\mathbf{p}}_{R_{k'|k'}} \\ \mathbf{0} & 1 \\ \mathbf{I} & \mathbf{J}\hat{\mathbf{p}}_{L_{1|k'}} \\ \vdots & \vdots \\ \mathbf{I} & \mathbf{J}\hat{\mathbf{p}}_{L_{M'|k'}} \end{bmatrix} \tag{37}$$

Now let us consider the case of standard SWF-SLAM. We first notice that (34) can be equivalently re-written as:

$$\begin{aligned} \mathbf{A}(k') = & \sum_{\mathbf{z}_{ij}\in\mathcal{Z}_M} \mathbf{H}_{ij}^T(k)\mathbf{R}_{ij}^{-1}\mathbf{H}_{ij}(k) + \sum_{\mathbf{z}_{ij}\in\mathcal{Z}_A} \mathbf{H}_{ij}^T(k')\mathbf{R}_{ij}^{-1}\mathbf{H}_{ij}(k') + \\ & \sum_{\kappa=1}^{m-1} \mathbf{\Phi}_\kappa^T(k)\mathbf{Q}_\kappa'^{-1}\mathbf{\Phi}_\kappa(k) + \sum_{\kappa=m}^{k'} \mathbf{\Phi}_\kappa^T(k')\mathbf{Q}_\kappa'^{-1}\mathbf{\Phi}_\kappa(k') \end{aligned} \tag{38}$$

It is important to note that now all the Jacobians involving the marginalized state are evaluated using the measurements up to time-step k, i.e., using the state estimates $\hat{\mathbf{x}}_{0:k|k}$, while the Jacobians involving the new states are computed using the latest estimates $\hat{\mathbf{x}}_{0:k'|k'}$. To see the structure more clearly, we can partition $\mathbf{A}(k')$ according to the state partitioning as follows:

$$\begin{aligned} \mathbf{A}(k') &= \underbrace{\begin{bmatrix} \mathbf{A}_{mm}(k) & \mathbf{A}_{mr}(k) & \mathbf{0} \\ \mathbf{A}_{rm}(k) & \mathbf{A}_{rr}(k) & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix}}_{\mathbf{A}_1(k)} + \underbrace{\begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_{rr}(k') & \mathbf{A}_{rn}(k') \\ \mathbf{0} & \mathbf{A}_{nr}(k') & \mathbf{A}_{nn}(k') \end{bmatrix}}_{\mathbf{A}_2(k')} \\ &= \begin{bmatrix} \mathbf{A}_{mm}(k) & \mathbf{A}_{mr}(k) & \mathbf{0} \\ \mathbf{A}_{rm}(k) & \mathbf{A}_{rr}(k) + \mathbf{A}_{rr}(k') & \mathbf{A}_{rn}(k') \\ \mathbf{0} & \mathbf{A}_{nr}(k') & \mathbf{A}_{nn}(k') \end{bmatrix} \end{aligned} \tag{39}$$

It is not difficult to verify that since different estimates, $\hat{\mathbf{x}}_{\mathbf{R}_{k|k}}$ and $\hat{\mathbf{x}}_{\mathbf{R}_{k'|k'}}$, are used in computing the Hessian matrix of the standard SWF, $\mathbf{A}(k')$ will have a nullspace of *lower* dimension than $\mathbf{A}_b(k')$. This implies that the standard SWF will acquire spurious information from the available measurements and thus can become inconsistent.

3 Observability Constrained (OC)-SWF

As seen from the preceding section, due to marginalization, the standard SWF possesses different parameter observability properties from the batch-MAP estimator, since its Hessian has a nullspace of lower dimension than that of the batch-MAP estimator. This implies that the standard SWF acquires spurious information along one direction of the state space (the one corresponding to global orientation), which can lead to inconsistency. To address this issue, we adopt the idea of observability-based rules for choosing linearization points that was originally proposed in our previous work [9], and develop a new observability constrained (OC)-SWF.

The key idea of the proposed approach is that the linearization points used in computing the Hessian matrix are selected so as to ensure that the Hessian has a nullspace of the same dimension as that of the batch MAP estimator (see (37)). Different approaches for selecting linearization points are possible to satisfy this observability condition. For example, the prior-linearization (PL)-SWF proposed in [7] employs a simple linearization scheme to achieve this goal based on [12]. Specifically, when computing the Hessian, it uses the prior estimates, $\hat{\mathbf{x}}_{\mathbf{R}}(k)$, instead of the current estimates $\hat{\mathbf{x}}_{\mathbf{R}}(k')$, for the states in $\mathbf{x}_{\mathbf{R}}$ that are connected to

marginalized states. By doing so, it is guaranteed that the same estimate is used as the linearization point for each of these states. However, even though the PL-SWF typically performs substantially better than the standard SWF, the prior estimates $\hat{\mathbf{x}}_{\mathbf{R}}(k)$ used as linearization points could be inaccurate, and thus can result in large linearization errors, which can degrade the estimator's performance. Therefore, in the proposed OC-SWF, we select the linearization points for the states $\mathbf{x}_{\mathbf{R}}$ and $\mathbf{x}_{\mathbf{N}}$ (i.e., the states that are still "active" in the minimization), in a way that not only ensures the correct dimension for the nullspace of the Hessian matrix, but also minimizes their difference from the current best available estimates (see [9]). This can be formulated as the following constrained minimization problem:[2]

$$\min_{\mathbf{x}^\star_{\mathbf{R}},\mathbf{x}^\star_{\mathbf{N}}} \quad ||\mathbf{x}^\star_{\mathbf{R}} - \hat{\mathbf{x}}_{\mathbf{R}}(k')||^2 + ||\mathbf{x}^\star_{\mathbf{N}} - \hat{\mathbf{x}}_{\mathbf{N}}(k')||^2 \tag{40}$$

$$\text{subject to} \quad \mathbf{A}(k')\mathbf{N}(k') = \mathbf{0} \tag{41}$$

In this formulation, $\mathbf{N}(k')$ is a design choice that defines the desired nullspace with correct dimension. Ideally, we would like to have the same nullspace as (37). However, this is not possible, as in the SWF some of the old states have been marginalized, and thus we do not maintain estimates for them. We next describe our choice of estimates used for constructing $\mathbf{N}(k')$, and denote these estimates by the symbol " $\bar{}$ ". Specifically, during the $(\ell+1)$-th Gauss-Newton iteration, we use the following estimates to construct the matrix $\mathbf{N}(k')$: (i) For the new states, $\mathbf{x}_{\mathbf{N}}$, as well as those states in $\mathbf{x}_{\mathbf{R}}$ for which no prior exists, we use the estimates from the ℓ-th iteration, i.e., $\bar{\mathbf{x}}_i = \hat{\mathbf{x}}_i(k')$; (ii) For all marginalized states, $\mathbf{x}_{\mathbf{M}}$, as well as for states in $\mathbf{x}_{\mathbf{R}}$ for which a prior exists, we use the prior estimate, i.e., $\bar{\mathbf{x}}_i = \hat{\mathbf{x}}_i(k)$. By replacing the pertinent state estimates in (37) by the estimates selected above, $\bar{\mathbf{x}}_{0:k'} = \begin{bmatrix}\bar{\mathbf{x}}^T_{\mathbf{M}} & \bar{\mathbf{x}}^T_{\mathbf{R}} & \bar{\mathbf{x}}^T_{\mathbf{N}}\end{bmatrix}^T$, we obtain the desired nullspace,

[2]For the clarity of presentation, hereafter the superscript (ℓ) is dropped, since, without loss of generality, we consider the $(\ell+1)$-th iteration in Gauss-Newton given that the results from the ℓ-th iteration are available.

$\mathbf{N}(k') = \mathbf{N}(\bar{\mathbf{x}}_{0:k'})$, as follows:

$$\mathbf{N}(k') \triangleq \begin{bmatrix} \mathbf{I} & \mathbf{J}\hat{\mathbf{p}}_{R_{0|k}} \\ \mathbf{0} & 1 \\ \vdots & \vdots \\ \mathbf{I} & \mathbf{J}\hat{\mathbf{p}}_{R_{m|k}} \\ \mathbf{0} & 1 \\ \mathbf{I} & \mathbf{J}\hat{\mathbf{p}}_{L_{M_1|k}} \\ \vdots & \vdots \\ \mathbf{I} & \mathbf{J}\hat{\mathbf{p}}_{L_{M_m|k}} \\ \cdots\cdots & \cdots\cdots \\ \mathbf{I} & \mathbf{J}\hat{\mathbf{p}}_{R_{Rm_1|k}} \\ \mathbf{0} & 1 \\ \vdots & \vdots \\ \mathbf{I} & \mathbf{J}\hat{\mathbf{p}}_{R_{Rm_m|k}} \\ \mathbf{0} & 1 \\ \mathbf{I} & \mathbf{J}\hat{\mathbf{p}}_{L_{Rm_1|k}} \\ \vdots & \vdots \\ \mathbf{I} & \mathbf{J}\hat{\mathbf{p}}_{L_{Rm_r|k}} \\ \cdots\cdots & \cdots\cdots \\ \mathbf{I} & \mathbf{J}\hat{\mathbf{p}}_{R_{Rn_1|k'}} \\ \mathbf{0} & 1 \\ \vdots & \vdots \\ \mathbf{I} & \mathbf{J}\hat{\mathbf{p}}_{R_{Rn_m|k'}} \\ \mathbf{0} & 1 \\ \mathbf{I} & \mathbf{J}\hat{\mathbf{p}}_{L_{Rn_1|k'}} \\ \vdots & \vdots \\ \mathbf{I} & \mathbf{J}\hat{\mathbf{p}}_{L_{Rn_r|k'}} \\ \cdots\cdots & \cdots\cdots \\ \mathbf{I} & \mathbf{J}\hat{\mathbf{p}}_{R_{k+1|k'}} \\ \mathbf{0} & 1 \\ \vdots & \vdots \\ \mathbf{I} & \mathbf{J}\hat{\mathbf{p}}_{R_{k'|k'}} \\ \mathbf{0} & 1 \\ \mathbf{I} & \mathbf{J}\hat{\mathbf{p}}_{L_{N_1|k'}} \\ \vdots & \vdots \\ \mathbf{I} & \mathbf{J}\hat{\mathbf{p}}_{L_{N_n|k'}} \end{bmatrix} =: \begin{bmatrix} \mathbf{N}_m(k) \\ \mathbf{N}_r(k) \\ \mathbf{N}_r(k') \\ \mathbf{N}_n(k') \end{bmatrix} \tag{42}$$

Once the design choice $\mathbf{N}_{k'}$ is made, we next introduce a novel approach to find optimal linearization points in computing Jacobians and thus computing Hessians in Newton-Gauss method, by solving a constrained optimization problem. It should be noted that by construction, $\mathbf{A}_1(k)\mathbf{N}(k') = \mathbf{0}$ (see (39), (37), and (42)). Thus, the above problem (40)-(41) can be equivalently written as:

$$\min \quad ||\mathbf{x}^\star_{\mathbf{R}} - \hat{\mathbf{x}}_{\mathbf{R}}(k')||^2 + ||\mathbf{x}^\star_{\mathbf{N}} - \hat{\mathbf{x}}_{\mathbf{N}}(k')||^2 \tag{43}$$

$$\text{subject to} \quad \mathbf{A}_2(k')\mathbf{N}(k') = \left(\sum_{\mathbf{z}_{ij} \in \mathcal{Z}_A} \mathbf{H}^T_{ij}(k')\mathbf{R}^{-1}_{ij}\mathbf{H}_{ij}(k') + \sum_{\kappa=m}^{k'} \mathbf{\Phi}^T_\kappa(k')\mathbf{Q}'^{-1}_\kappa \mathbf{\Phi}_\kappa(k') \right) \mathbf{N}(k') = \mathbf{0} \tag{44}$$

Now we present an analytical solution to this problem. Specifically, the constraint (44) is equivalent to:

$$\left(\sum_{\mathbf{z}_{ij} \in \mathcal{Z}_A} \mathbf{H}_{ij}^T(k') \mathbf{R}_{ij}^{-1} \mathbf{H}_{ij}(k') + \sum_{\kappa=k}^{k'} \mathbf{\Phi}_\kappa^T(k') \mathbf{Q}_\kappa'^{-1} \mathbf{\Phi}_\kappa(k') \right) \mathbf{N}(k') = \mathbf{0} \tag{45}$$

$$\Rightarrow \quad \begin{bmatrix} \cdots & \mathbf{H}_{ij}^T & \cdots & \mathbf{\Phi}_k^T & \cdots & \mathbf{\Phi}_{k'}^T \end{bmatrix} \begin{bmatrix} \ddots & & & & & \\ & \mathbf{R}_{ij} & & & & \\ & & \ddots & & & \\ & & & \mathbf{Q}_k' & & \\ & & & & \ddots & \\ & & & & & \mathbf{Q}_{k'}' \end{bmatrix} \begin{bmatrix} \vdots \\ \mathbf{H}_{ij} \\ \vdots \\ \mathbf{\Phi}_k \\ \vdots \\ \mathbf{\Phi}_{k'} \end{bmatrix} \mathbf{N}(k') = \mathbf{0} \tag{46}$$

$$\Rightarrow \quad \begin{bmatrix} \vdots \\ \mathbf{H}_{ij} \\ \vdots \\ \mathbf{\Phi}_k \\ \vdots \\ \mathbf{\Phi}_{k'} \end{bmatrix} \mathbf{N}(k') = \mathbf{0} \tag{47}$$

$$\Rightarrow \quad \begin{cases} \mathbf{H}_{ij} \mathbf{N}(k') = \mathbf{0} \ , & \forall \mathbf{z}_{ij} \in \mathcal{Z}_A \\ \mathbf{\Phi}_\kappa \mathbf{N}(k') = \mathbf{0} \ , & \forall \kappa = m, \ldots, k' \end{cases} \tag{48}$$

Note that the measurement Jacobian, $\mathbf{H}_{ij}$, is spare and only depends on the landmark position and the robot pose at the time when observing the landmark (see (9)). $\mathcal{Z}_A$ only contains the measurements involving the remaining state $\mathbf{x_R}$ and the new state $\mathbf{x_N}$. So, the above constraints can be written as (without loss of generality, assuming that the robot observes landmark i at time-step j):

$$\mathbf{H}_{ij} \mathbf{N}(k') = \mathbf{0} \tag{49}$$

$$\Rightarrow \quad (\nabla \mathbf{h}_{ij}) \mathbf{C}^T(\phi_{R_j}^\star) \begin{bmatrix} -\mathbf{I}_2 & -\mathbf{J}(\mathbf{p}_{L_i}^\star - \mathbf{p}_{R_j}^\star) & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{J}\bar{\mathbf{p}}_{R_j} \\ \mathbf{0} & 1 \\ \mathbf{I} & \mathbf{J}\bar{\mathbf{p}}_{L_i} \end{bmatrix} = \mathbf{0} \tag{50}$$

$$\Rightarrow \quad \mathbf{p}_{R_j}^\star - \bar{\mathbf{p}}_{R_j} + \bar{\mathbf{p}}_{L_i} - \mathbf{p}_{L_i}^\star = \mathbf{0} \tag{51}$$

Similarly,

$$\mathbf{\Phi}_\kappa \mathbf{N}(k') = \mathbf{0} \tag{52}$$

$$\Rightarrow \quad \begin{bmatrix} -\mathbf{I}_2 & -\mathbf{J}\left(\mathbf{p}_{R_{\kappa+1}}^\star - \mathbf{p}_{R_\kappa}^\star\right) & \mathbf{I} & \mathbf{0} \\ \mathbf{0}_{1\times 2} & -1 & \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{J}\bar{\mathbf{p}}_{R_\kappa} \\ \mathbf{0} & 1 \\ \mathbf{I} & \mathbf{J}\bar{\mathbf{p}}_{R_{\kappa+1}} \\ \mathbf{0} & 1 \end{bmatrix} = \mathbf{0} \tag{53}$$

$$\Rightarrow \quad \mathbf{p}_{R_\kappa}^\star - \bar{\mathbf{p}}_{R_\kappa} + \bar{\mathbf{p}}_{R_{\kappa+1}} - \mathbf{p}_{R_{\kappa+1}}^\star = \mathbf{0} \tag{54}$$

Therefore, using the results of (51) and (54), the problem (40)-(41) can be simplified as:

$$\min_{\mathbf{x}_\mathbf{R}^\star, \mathbf{x}_\mathbf{N}^\star} \quad ||\mathbf{x}_\mathbf{R}^\star - \hat{\mathbf{x}}_\mathbf{R}(k')||^2 + ||\mathbf{x}_\mathbf{N}^\star - \hat{\mathbf{x}}_\mathbf{N}(k')||^2 \tag{55}$$

$$\text{s.t.} \begin{cases} \mathbf{p}_{R_\kappa}^\star - \bar{\mathbf{p}}_{R_\kappa} + \bar{\mathbf{p}}_{R_{\kappa+1}} - \mathbf{p}_{R_{\kappa+1}}^\star = \mathbf{0} \ , & \forall \kappa = m, \ldots, k' \\ \mathbf{p}_{R_j}^\star - \bar{\mathbf{p}}_{R_j} + \bar{\mathbf{p}}_{L_i} - \mathbf{p}_{L_i}^\star = \mathbf{0} \ , & \forall \mathbf{z}_{ij} \in \mathcal{Z}_\mathbf{A} \end{cases} \tag{56}$$

We now derive an analytical solution to the constrained minimization problem (55)-(56). In particular, the approach of Lagrangian multipliers [13] is employed. The Lagrangian function is constructed as follows:

$$\begin{aligned}\mathcal{L} = &||\mathbf{x}^{\star}_{\mathbf{R}} - \hat{\mathbf{x}}_{\mathbf{R}}(k')||^2 + ||\mathbf{x}^{\star}_{\mathbf{N}} - \hat{\mathbf{x}}_{\mathbf{N}}(k')||^2 \\ &+ \sum_{\kappa=m}^{k'-1} \boldsymbol{\mu}_{\kappa}^T \left(\mathbf{p}^{\star}_{R_\kappa} - \bar{\mathbf{p}}_{R_\kappa} + \bar{\mathbf{p}}_{R_{\kappa+1}} - \mathbf{p}^{\star}_{R_{\kappa+1}}\right) \\ &+ \sum_{(i,j),\mathbf{z}_{ij}\in\mathcal{Z}_A} \boldsymbol{\lambda}_{ij}^T \left(\mathbf{p}^{\star}_{R_j} - \bar{\mathbf{p}}_{R_j} + \bar{\mathbf{p}}_{L_i} - \mathbf{p}^{\star}_{L_i}\right)\end{aligned} \tag{57}$$

By setting the derivatives with respect to the state and Lagrangian-multiplier variables equal to zero, we have:

$$\frac{\partial \mathcal{L}}{\partial \mathbf{p}^{\star}_{R_\kappa}} = \begin{cases} 2(\mathbf{p}^{\star}_{R_\kappa} - \hat{\mathbf{p}}_{R_{\kappa|k}}) + \boldsymbol{\mu}_{\kappa} + \sum\limits_{i,\mathbf{z}_{i\kappa}\in\mathcal{Z}_{\mathbf{A}}} \boldsymbol{\lambda}_{i\kappa} = \mathbf{0}\ , & \text{if } \kappa = m \\ 2(\mathbf{p}^{\star}_{R_\kappa} - \hat{\mathbf{p}}_{R_{\kappa|k}}) - \boldsymbol{\mu}_{\kappa-1} + \sum\limits_{i,\mathbf{z}_{i\kappa}\in\mathcal{Z}_{\mathbf{A}}} \boldsymbol{\lambda}_{i\kappa} = \mathbf{0}\ , & \text{if } \kappa = k' \\ 2(\mathbf{p}^{\star}_{R_\kappa} - \hat{\mathbf{p}}_{R_{\kappa|k}}) + \boldsymbol{\mu}_{\kappa} - \boldsymbol{\mu}_{\kappa-1} + \sum\limits_{i,\mathbf{z}_{i\kappa}\in\mathcal{Z}_{\mathbf{A}}} \boldsymbol{\lambda}_{i\kappa} = \mathbf{0}\ , & \text{else} \end{cases} \tag{58}$$

$$\frac{\partial \mathcal{L}}{\partial \mathbf{p}^{\star}_{L_i}} = 2(\mathbf{p}^{\star}_{L_i} - \hat{\mathbf{p}}_{L_{i|k}}) - \sum_{j,\mathbf{z}_{ij}\in\mathcal{Z}_{\mathbf{A}}} \boldsymbol{\lambda}_{ij} = \mathbf{0} \tag{59}$$

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{\mu}_{\kappa}} = \mathbf{p}^{\star}_{R_\kappa} - \bar{\mathbf{p}}_{R_\kappa} + \bar{\mathbf{p}}_{R_{\kappa+1}} - \mathbf{p}^{\star}_{R_{\kappa+1}} = \mathbf{0} \tag{60}$$

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{\lambda}_{ij}} = \mathbf{p}^{\star}_{R_j} - \bar{\mathbf{p}}_{R_j} + \bar{\mathbf{p}}_{L_i} - \mathbf{p}^{\star}_{L_i} = \mathbf{0} \tag{61}$$

$$\frac{\partial \mathcal{L}}{\partial \mathbf{x}^{\star}_{\text{other}}} = 2(\mathbf{x}^{\star}_{\text{other}} - \hat{\mathbf{x}}_{\text{other}}(k')) = \mathbf{0} \tag{62}$$

where $\mathbf{x}_{\text{other}}$ denotes all the state variables except the ones involved in (58)-(61). Solving (58), (59), and (62) yields the following optimal solutions:

$$\mathbf{p}^{\star}_{R_\kappa} = \hat{\mathbf{p}}_{R_{\kappa|k}} - \frac{1}{2}\left[\boldsymbol{\delta\mu}_{\kappa} + \sum_{i,\mathbf{z}_{i\kappa}\in\mathcal{Z}_{\mathbf{A}}} \boldsymbol{\lambda}_{i\kappa}\right] \tag{63}$$

$$\mathbf{p}^{\star}_{L_i} = \hat{\mathbf{p}}_{L_{i|k}} + \frac{1}{2}\left[\sum_{j,\mathbf{z}_{ij}\in\mathcal{Z}_{\mathbf{A}}} \boldsymbol{\lambda}_{ij}\right] \tag{64}$$

$$\mathbf{x}^{\star}_{\text{other}} = \hat{\mathbf{x}}_{\text{other}}(k') \tag{65}$$

where

$$\boldsymbol{\delta\mu}_{\kappa} = \begin{cases} \boldsymbol{\mu}_{\kappa}\ , & \text{if } \kappa = m \\ -\boldsymbol{\mu}_{\kappa-1}\ , & \text{if } \kappa = k' \\ \boldsymbol{\mu}_{\kappa} - \boldsymbol{\mu}_{\kappa-1}\ , & \text{else} \end{cases}$$

Substituting (63)-(65) into (60) and (61) yields the following *linear* equations in terms of the Lagrangian multipliers:

$$\boldsymbol{\Delta\mu}_{\kappa} + \sum_{i,\mathbf{z}_{i\kappa}\in\mathcal{Z}_A} \boldsymbol{\lambda}_{i\kappa} - \sum_{i,\mathbf{z}_{i(\kappa+1)}\in\mathcal{Z}_A} \boldsymbol{\lambda}_{i(\kappa+1)} = 2\left(\hat{\mathbf{p}}_{R_{\kappa|k}} - \bar{\mathbf{p}}_{R_\kappa} + \bar{\mathbf{p}}_{R_{\kappa+1}} - \hat{\mathbf{p}}_{R_{\kappa+1|k}}\right) \tag{66}$$

$$\boldsymbol{\delta\mu}_{\kappa} + \sum_{i,\mathbf{z}_{i\kappa}\in\mathcal{Z}_A} \boldsymbol{\lambda}_{i\kappa} + \sum_{j,\mathbf{z}_{ij}\in\mathcal{Z}_A} \boldsymbol{\lambda}_{ij} = 2\left(\hat{\mathbf{p}}_{R_{\kappa|k}} - \bar{\mathbf{p}}_{R_\kappa} + \bar{\mathbf{p}}_{L_i} - \hat{\mathbf{p}}_{L_{i|k}}\right) \tag{67}$$

where

$$\boldsymbol{\Delta\mu}_\kappa = \begin{cases} 2\boldsymbol{\mu}_\kappa - \boldsymbol{\mu}_{\kappa+1} \ , & \text{if } \kappa = m \\ 2\boldsymbol{\mu}_\kappa - \boldsymbol{\mu}_{\kappa-1} \ , & \text{if } \kappa = k'-1 \\ -\boldsymbol{\mu}_{\kappa-1} \ , & \text{if } \kappa = k' \\ 2\boldsymbol{\mu}_\kappa - \boldsymbol{\mu}_{\kappa-1} - \boldsymbol{\mu}_{\kappa+1} \ , & \text{else} \end{cases}$$

In order to determine the Lagrangian multipliers, $\boldsymbol{\mu}_\kappa$ and $\boldsymbol{\lambda}_{ij}$, we stack equations (66)-(67) for all the measurements (constraints) into matrix-vector form and solve the resulting linear system. Once the Lagrangian multipliers are specified, the *optimal* linearization points can be obtained based on (63)-(65). Subsequently, the Jacobian and Hessian matrices are computed using the optimal linearization points, and then the standard Gauss-Newton steps are carried out (see Section 2.1). It should be pointed out that, as compared to the standard SWF and the PL-SWF, the OC-SWF only requires an additional computational overhead of linearly solving for the Lagrangian multipliers, which in general is cubic in the number of active proprioceptive and exteroceptive measurements.

References

[1] S. Kay, *Fundamentals of Statistical Signal Processing, Vol. I - Estimation Theory*. Prentice Hall, 1993.

[2] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon, "Bundle adjustment – a modern synthesis," *Lecture Notes in Computer Science*, vol. 1883, pp. 298–375, Jan. 2000.

[3] 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.

[4] G. Sibley, L. Matthies, and G. Sukhatme, "Sliding window filter with application to planetary landing," *Journal of Field Robotics*, vol. 27, no. 5, pp. 587–608, Sep./Oct. 2010.

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

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

[7] T. Dong-Si and A. I. Mourikis, "Motion tracking with fixed-lag smoothing: Algorithm and consistency analysis," in *Proc. IEEE International Conference on Robotics and Automation*, Shanghai, China, May 9–13, 2011, pp. 5655–5662.

[8] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with applications to tracking and navigation*. New York: Wiley, 2001.

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

[10] 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.

[11] A. Mourikis, N. Trawny, S. Roumeliotis, A. Johnson, A. Ansar, and L. Matthies, "Vision-aided inertial navigation for spacecraft entry, descent, and landing," *IEEE Transactions on Robotics*, vol. 25, no. 2, pp. 264 –280, Apr. 2009.

[12] G. P. Huang, A. I. Mourikis, and S. I. Roumeliotis, "Analysis and improvement of the consistency of extended Kalman filter-based SLAM," in *Proc. IEEE International Conference on Robotics and Automation*, Pasadena, CA, May 19–23, 2008, pp. 473–479.

[13] D. P. Bertsekas, *Nonlinear Programming*. Athena Scientific, 1999.