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: tdong002@student.ucr.edu, mourikis@ee.ucr.edu

Updated: October 11, 2010

Abstract

This report 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.

Contents

1 Introduction and Relation to Prior Work

In this work, we focus on the problem of *motion tracking* by combining a robot's proprioceptive and exteroceptive sensor measurements. The simplest approach for tracking the trajectory of a robot is *dead reckoning*, which consists of integrating proprioceptive (e.g., odometry) measurements. However, in most practical cases dead reckoning results in rapid accumulation of uncertainty, which quickly renders the estimates unusable. To reduce the rate at which the uncertainty increases, we can utilize the measurements from the robot's exteroceptive sensors (e.g., camera, laser scanner). By detecting static features in consecutive time instants, the robot can extract useful information about its ego-motion, and utilize it to improve its pose estimates. We note that in this work, we concentrate on the task of utilizing the *local* motion information provided by the robot's sensors, i.e., we do not address the issue of loop closing.

The processing of exteroceptive sensor measurements for motion tracking has attracted considerable research interest, primarily in the context of vision-based motion estimation (*visual odometry* (VO) [1]). A large class of techniques utilize the feature measurements to derive constraints between *pairs* of consecutive robot poses (e.g., [1–5] and references therein). However, when a feature is observed from multiple robot poses, pair-wise processing inevitably results in loss of information. In order to properly utilize the motion information in the (typical) case where features are tracked over multiple consecutive time instants, pairwise processing is not appropriate. Instead, we must maintain estimates of a *sliding window* of poses, since this enables us to process all the relative-motion constraints between these poses. Methods that estimate the state over a sliding window of time are commonly called *fixed-lag smoothing* algorithms [6].

While extended Kalman filtering (EKF)-based fixed-lag smoothing approaches to motion estimation have appeared in the literature (e.g., [7]), 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 landmarks, 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 poses are discarded is not optimal. Specifically, these older poses are assumed to be fixed, and are used to "bootstrap" the trajectory estimates. This approximate approach ignores the uncertainty of the older poses, 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 un-observable 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]).

2 Fixed-lag smoothing

2.1 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 report. The list of most important notations used is shown in Table 1 for reference.

At a 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 the equation

$$\mathbf{r}_{i+1} = \mathbf{f}(\mathbf{r}_i, \mathbf{u}_i) + \mathbf{w}_i, \tag{1}$$

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:

$$\mathbf{z}_{ij} = \mathbf{h}(\mathbf{r}_i, \mathbf{l}_j) + \mathbf{n}_{ij} \tag{2}$$

where $\mathbf{n}_{ij}$ is the measurement noise vector, assumed to be zero-mean, white, and Gaussian, with covariance matrix $\mathbf{R}_{ij}$. Each of the measurements is of dimension d_m .

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 [18]:

$$c(\mathbf{x}_k) = \frac{1}{2}||\mathbf{r}_0 - \hat{\mathbf{r}}_{p_0}||_{\mathbf{R}_p(0)} + \frac{1}{2} \sum_{(i,j)\in\mathcal{S}_a(k)} \gamma_{h_{ij}} + \frac{1}{2} \sum_{i=0}^{k-1} \gamma_{f_i} \tag{3}$$

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

$$\gamma_{h_{ij}} = ||\mathbf{z}_{ij} - \mathbf{h}(\mathbf{r}_i, \mathbf{l}_j)||_{\mathbf{R}_{ij}}, \tag{4}$$

$$\gamma_{f_i} = ||\mathbf{r}_{i+1} - \mathbf{f}(\mathbf{r}_i, \mathbf{u}_i)||_{\mathbf{Q}_i} \tag{5}$$

with the notation $||\mathbf{a}||_{\mathbf{M}} = \mathbf{a}^T\mathbf{M}^{-1}\mathbf{a}$. The size of the state vector $\mathbf{x}_k$ is denoted as d_{x_k}.

$c(\mathbf{x}_k)$ 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:

$$c(\mathbf{x}_k^{(\ell)} + \Delta\mathbf{x}) \simeq c(\mathbf{x}^{(\ell)}) + \mathbf{b}^{(\ell)T}\Delta\mathbf{x} + \frac{1}{2}\Delta\mathbf{x}^T\mathbf{A}^{(\ell)}\Delta\mathbf{x} \tag{6}$$

where $\mathbf{b}^{(\ell)} = \nabla c(\mathbf{x}_k^{(\ell)})$ and $\mathbf{A}^{(\ell)} = \nabla^2 c(\mathbf{x}_k^{(\ell)})$ denote the Jacobian and Hessian of c with respect to $\mathbf{x}_k$, evaluated at the current iteration, $\mathbf{x}_k^{(\ell)}$. Specifically, $\mathbf{b}^{(\ell)}$ is given by the $d_{x_k} \times 1$ vector:

$$\mathbf{b}^{(\ell)} = \mathbf{b}_0^p(\mathbf{x}_k^{(\ell)}) + \mathbf{b}_{\mathcal{S}_a(k)}^h(\mathbf{x}_k^{(\ell)}) + \mathbf{b}_{0:k}^f(\mathbf{x}_k^{(\ell)}) \tag{7}$$

where the three terms appearing in the sum 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:

$$\mathbf{b}_0^p(\mathbf{x}_k^{(\ell)}) = \mathbf{\Pi}_{d_r}^T \mathbf{A}_p(0)(\mathbf{r}_0^{(\ell)} - \hat{\mathbf{r}}_{p_0}), \tag{8}$$

[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.

Table 1: List of notations, in alphabetical order

Symbols	Definition	Defined on pg.
$\mathbf{A}$	Hessian of the cost function c with respect to $\mathbf{x}_k$	4
$\mathbf{G}_i$	process Jacobian, i.e. the Jacobian of $\mathbf{g}_i = \mathbf{r}_{i+1} - \mathbf{f}(\mathbf{r}_i, \mathbf{u}_i)$ with respect to $\mathbf{x}_k$	6
$\mathbf{H}_{ij}$	measurement Jacobian, i.e. the Jacobian of $\mathbf{h}(\mathbf{r}_i, \mathbf{l}_j)$ with respect to $\mathbf{x}_k$	6
$\mathbf{H}_{L_{ij}}$	Jacobian of $\mathbf{h}$ with respect to the landmark position	6
$\mathbf{H}_{R_{ij}}$	Jacobian of $\mathbf{h}$ with respect to the robot pose	6
$\mathbf{Q}_i$	covariance matrix of process noise	4
$\mathbf{R}_p(0)$	covariance matrix of prior for initial robot pose	4
$\mathbf{R}_{ij}$	covariance matrix of measurement noise	4
$\mathbf{b}$	Jacobian of the cost function c with respect to the state vector	4
$\mathbf{f}$	robot motion model function, $\mathbf{r}_{i+1} = \mathbf{f}(\mathbf{r}_i, \mathbf{u}_i) + \mathbf{w}_i$	4
$\mathbf{h}$	measurement model function, $\mathbf{z}_{ij} = \mathbf{h}(\mathbf{r}_i, \mathbf{l}_j) + \mathbf{n}_{ij}$	4
$\mathbf{l}_j$	landmark pose vector, which usually is the landmark position	4
$\mathbf{n}_{ij}$	measurement noise vector	4
$\mathbf{r}_i$	robot state vector at time step i, which may comprise position, orientation, velocity, etc.	4
$\hat{\mathbf{r}}_{p_0}$	prior estimate for initial robot pose	4
$\mathbf{u}_i$	measured control input vector (e.g., odometry)	4
$\mathbf{w}_i$	process noise vector	4
$\mathbf{x}_k$	state vector for the fixed-lag smoother at time step k, containing the states $\{\mathbf{r}_{0:k}, \mathbf{l}_{1:n}\}$	4
$\mathbf{x}_m$	vector of marginalized states $\{\mathbf{r}_0, \ldots, \mathbf{r}_{m-1}, \mathbf{l}_1, \ldots, \mathbf{l}_{m_l}\}$	6
$\mathbf{x}_r$	vector of states remaining after marginalization $\{\mathbf{r}_m, \ldots, \mathbf{r}_k, \mathbf{l}_{m_l+1}, \ldots, \mathbf{l}_N\}$	6
$\mathbf{x}_n$	vector of new states after marginalization $\{\mathbf{r}_{k+1}, \ldots, \mathbf{r}_{k'}, \mathbf{l}_{N+1}, \ldots, \mathbf{l}_{N+N_o}\}$	7
$\mathbf{z}_{ij}$	measurement vector, observation of landmark j from robot pose i	4
d_ℓ	dimension of landmark position (e.g. 2 in 2D)	4
d_m	dimension of measurement observation	4
d_r	dimension of robot pose (e.g. 3 in 2D)	4
d_{x_k}	dimension of state vector $\mathbf{x}_k$	4
i	index variable for robot poses	4
j	index variable for landmarks	4
k	time step	4
l	total number of measurements	11
l_i	number of landmarks observed at time step i	11
m_l	number of marginalized landmarks $\mathbf{l}_1, \ldots, \mathbf{l}_{m_l}$	6
m	number of marginalized robot poses $\mathbf{r}_0, \ldots, \mathbf{r}_{m_r-1}$	6
n	number of landmarks at time step k	4
n'	number of landmarks at time step k'	7
$\mathcal{S}_a$	set of pairs (i, j) indexing the measurements $\mathbf{z}_{ij}$ that involve currently active states	4
$\mathcal{S}_m$	set of pairs (i, j) indexing the measurements $\mathbf{z}_{ij}$ that involve marginalized states	7
$\mathcal{S}$	set of pairs (i, j) indexing all measurements, $\mathcal{S}_m \bigcup \mathcal{S}_a(k')$	10
$\mathbf{\Lambda}^f_{0:k}$	information matrix due to the process model for the time interval $[0, k]$	6
$\mathbf{\Lambda}^h_{\mathcal{S}_a(k)}$	information matrix due to the measurements indexed by $\mathcal{S}_a(k)$	6
$\mathbf{\Lambda}^p_0$	information matrix due to the prior at time 0	6
$\mathbf{\Phi}_i$	state transition matrix, Jacobian of $\mathbf{f}(\mathbf{r}_i, \mathbf{u}_i)$ with respect to $\mathbf{r}_i$	6

$$\mathbf{b}^h_{\mathcal{S}_a(k)}(\mathbf{x}_k^{(\ell)}) = -\sum_{(i,j)\in\mathcal{S}_a(k)} \mathbf{H}_{ij}^{(\ell)T} \mathbf{R}_{ij}^{-1} \left(\mathbf{z}_{ij} - \mathbf{h}(\mathbf{r}_i^{(\ell)}, \mathbf{l}_j^{(\ell)})\right) \tag{9}$$

$$\mathbf{b}^f_{0:k}(\mathbf{x}_k^{(\ell)}) = \sum_{i=0}^{k-1} \mathbf{G}_i^{(\ell)T} \mathbf{Q}_i^{-1} \left(\mathbf{r}_{i+1}^{(\ell)} - \mathbf{f}(\mathbf{r}_i^{(\ell)}, \mathbf{u}_i)\right) \tag{10}$$

In these expressions we denote $\mathbf{A}_p(0) = \mathbf{R}_p(0)^{-1}$, and

$$\mathbf{\Pi}_{d_r} = \begin{bmatrix} \mathbf{I}_{d_r} & \mathbf{0}_{d_r \times (d_{x_k} - d_r)} \end{bmatrix} \tag{11}$$

with $\mathbf{I}_n$ being the $n \times n$ 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 the current iterate $\mathbf{x}_k^{(\ell)}$. 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, $\mathbf{H}_{ij}^{(\ell)}$ is the $d_m \times d_{x_k}$ matrix:

$$\mathbf{H}_{ij}^{(\ell)} = \begin{bmatrix} \mathbf{0} & \dots & \mathbf{H}_{R_{ij}}^{(\ell)} & \dots & \mathbf{H}_{L_{ij}}^{(\ell)} & \dots & \mathbf{0} \end{bmatrix} \tag{12}$$

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 $\mathbf{G}_i^{(\ell)}$ is the $d_r \times d_{x_k}$ matrix:

$$\mathbf{G}_i^{(\ell)} = \begin{bmatrix} \mathbf{0} & \dots & -\mathbf{\Phi}_i^{(\ell)} & \mathbf{I}_{d_r} & \dots & \mathbf{0} \end{bmatrix} \tag{13}$$

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]:

$$\mathbf{A}^{(\ell)} = \mathbf{\Lambda}_0^p + \mathbf{\Lambda}_{\mathcal{S}_a(k)}^h(\mathbf{x}_k^{(\ell)}) + \mathbf{\Lambda}_{0:k}^f(\mathbf{x}_k^{(\ell)}), \tag{14}$$

where

$$\mathbf{\Lambda}_0^p = \mathbf{\Pi}_{d_r}^T \mathbf{A}_p(0) \mathbf{\Pi}_{d_r} \tag{15}$$

$$\mathbf{\Lambda}_{\mathcal{S}_a(k)}^h(\mathbf{x}_k^{(\ell)}) = \sum_{(i,j) \in \mathcal{S}_a(k)} \mathbf{H}_{ij}^{(\ell)T} \mathbf{R}_{ij}^{-1} \mathbf{H}_{ij}^{(\ell)} \tag{16}$$

$$\mathbf{\Lambda}_{0:k}^f(\mathbf{x}_k^{(\ell)}) = \sum_{i=0}^{k-1} \mathbf{G}_i^{(\ell)T} \mathbf{Q}_i^{-1} \mathbf{G}_i^{(\ell)} \tag{17}$$

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 $\mathbf{x}$. All these matrices are of size $d_{x_k} \times d_{x_k}$.

The value of $\Delta\mathbf{x}^{(\ell)}$ minimizing the cost function (6) is found by solving the linear system:

$$\mathbf{A}^{(\ell)} \Delta\mathbf{x}^{(\ell)} = -\mathbf{b}^{(\ell)} \tag{18}$$

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

2.2 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 approximately constant). Therefore, in order to obtain an algorithm with approximately constant 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 defined in Eq. (3).

We consider the following scenario: The robot collects measurements during the time interval $[0, k]$, and a full-state MAP estimation step is carried out at time step k. Then, the states $\mathbf{x}_m = \{\mathbf{r}_0, \dots, \mathbf{r}_{m-1}, \mathbf{l}_1, \dots, \mathbf{l}_{m_l}\}$ (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 $\mathbf{x}_r = \{\mathbf{r}_m, \dots, \mathbf{r}_k, \mathbf{l}_{m_l+1}, \dots, \mathbf{l}_n\}$ 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 (3). This cost function has the special structure shown below:

$$c(\mathbf{x}_{k'}) = c(\mathbf{x}_m, \mathbf{x}_r, \mathbf{x}_n) = c_n(\mathbf{x}_r, \mathbf{x}_n) + c_m(\mathbf{x}_r, \mathbf{x}_m) \tag{19}$$

The cost 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 quadratic 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 values are *not* known, and therefore the above minimization needs to occur 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 (19), to employ the following property, which holds for any multi-variable optimization problem:

$$\min_{\mathbf{x}_m, \mathbf{x}_r, \mathbf{x}_n} c(\mathbf{x}_m, \mathbf{x}_r, \mathbf{x}_n) = \min_{\mathbf{x}_r, \mathbf{x}_n} \left(\min_{\mathbf{x}_m} c(\mathbf{x}_m, \mathbf{x}_r, \mathbf{x}_n) \right) = \min_{\mathbf{x}_r, \mathbf{x}_n} \left(c_n(\mathbf{x}_r, \mathbf{x}_n) + \min_{\mathbf{x}_m} c_m(\mathbf{x}_r, \mathbf{x}_m) \right) \tag{20}$$

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 exactly, and we once again employ the second-order Taylor-series approximation:

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

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

$$\mathbf{b}_m(k) = \mathbf{b}_0^p\big(\hat{\mathbf{x}}_r(k), \hat{\mathbf{x}}_m(k)\big) + \mathbf{b}_{\mathcal{S}_m}^h\big(\hat{\mathbf{x}}_r(k), \hat{\mathbf{x}}_m(k)\big) + \mathbf{b}_{0:m}^f\big(\hat{\mathbf{x}}_r(k), \hat{\mathbf{x}}_m(k)\big) \tag{21}$$

$$\mathbf{A}_m(k) = \mathbf{\Lambda}_0^p + \mathbf{\Lambda}_{\mathcal{S}_m}^h\big(\hat{\mathbf{x}}_r(k), \hat{\mathbf{x}}_m(k)\big) + \mathbf{\Lambda}_{0:m}^f\big(\hat{\mathbf{x}}_r(k), \hat{\mathbf{x}}_m(k)\big) \tag{22}$$

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

$$\mathbf{b}_m(k) = \begin{bmatrix} \mathbf{b}_{mm}(k) \\ \mathbf{b}_{mr}(k) \end{bmatrix} \quad \mathbf{A}_m(k) = \begin{bmatrix} \mathbf{A}_{mm}(k) & \mathbf{A}_{mr}(k) \\ \mathbf{A}_{rm}(k) & \mathbf{A}_{rr}(k) \end{bmatrix} \tag{23}$$

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 (21) is a quadratic function of $\mathbf{x}_m$, and its minimum with respect to $\mathbf{x}_m$ is attained for

$$\mathbf{x}_m = \hat{\mathbf{x}}_m(k) - \mathbf{A}_{mm}(k)^{-1}\big(\mathbf{b}_{mm}(k) + \mathbf{A}_{mr}(k)\big(\mathbf{x}_r - \hat{\mathbf{x}}_r(k)\big)\big)$$

Substitution in (21) yields the minimum value of $\mathbf{c}_m$ with respect to $\mathbf{x}_m$:

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

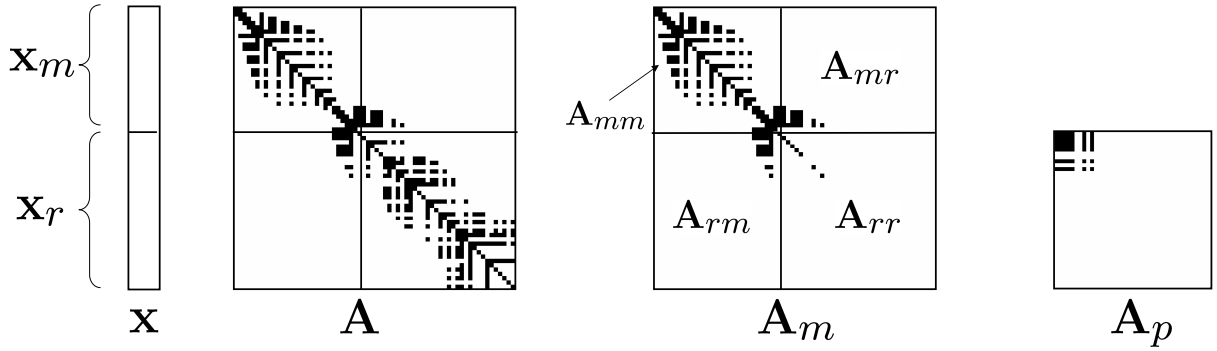


Figure 1: Illustration of the sparsity patterns in the matrices appearing in the fixed-lag smoothing algorithm. Here we employ a temporal ordering of the variables (i.e., robot poses and landmarks enter the state vector in the order they appear in time), although during the solution of the system, any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

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

$$\mathbf{b}_p(k) = \mathbf{b}_{mr}(k) - \mathbf{A}_{rm}(k)\mathbf{A}_{mm}(k)^{-1}\mathbf{b}_{mm}(k) \tag{24}$$

$$\mathbf{A}_p(k) = \mathbf{A}_{rr}(k) - \mathbf{A}_{rm}(k)\mathbf{A}_{mm}(k)^{-1}\mathbf{A}_{mr}(k) \tag{25}$$

Combining this result with that of (20), 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:

$$c'(\mathbf{x}_r, \mathbf{x}_n) = \mathbf{b}_p^T(k)(\mathbf{x}_r - \hat{\mathbf{x}}_r(k)) + \frac{1}{2}||\mathbf{x}_r - \hat{\mathbf{x}}_r(k)||_{\mathbf{A}_p(k)} + \frac{1}{2}\sum_{(i,j)\in\mathcal{S}_a(k')}\gamma_{h_{ij}} + \frac{1}{2}\sum_{i=m}^{k'-1}\gamma_{f_i} \tag{26}$$

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 (21). 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:

$$\mathbf{b}^{(\ell)} = \mathbf{b}_k^p(\mathbf{x}_r^{(\ell)}) + \mathbf{b}_{\mathcal{S}_a(k')}^h(\mathbf{x}_r^{(\ell)}, \mathbf{x}_n^{(\ell)}) + \mathbf{b}_{m:k'}^f(\mathbf{x}_r^{(\ell)}, \mathbf{x}_n^{(\ell)}) \tag{27}$$

$$\mathbf{A}^{(\ell)} = \mathbf{\Lambda}_k^p + \mathbf{\Lambda}_{\mathcal{S}_a(k')}^h(\mathbf{x}_r^{(\ell)}, \mathbf{x}_n^{(\ell)}) + \mathbf{\Lambda}_{m:k'}^f(\mathbf{x}_r^{(\ell)}, \mathbf{x}_n^{(\ell)}) \tag{28}$$

The terms in the above expressions are defined analogously to those in (7)-(14), with the exception of $\mathbf{b}_k^p(\mathbf{x}_r^{(\ell)})$, which is given by:

$$\mathbf{b}_k^p(\mathbf{x}_r^{(\ell)}) = \mathbf{\Pi}_r\mathbf{b}_p(k) + \mathbf{\Pi}_r\mathbf{A}_p(k)(\mathbf{x}_r^{(\ell)} - \hat{\mathbf{x}}_r(k)) \tag{29}$$

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

2.3 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

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 (29): $\mathbf{b}_p = \mathbf{0}$.

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 (27)-(29), respectively. Repeat until convergence.

Marginalization

- Set the new prior estimate as $\hat{\mathbf{x}}_p = \hat{\mathbf{x}}_r(k)$, and compute the new prior information matrix $\mathbf{A}_p$ and vector $\mathbf{b}_p$ via (24) and (25).
- Remove the states $\mathbf{x}_m$ from the active state vector, and discard all measurements that involve these states.

defined by (27) and (28). 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 $\mathbf{x}_m$, 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 (25), and the vector $\mathbf{b}_p(k)$ via (24). Once these quantities have been computed, the states $\mathbf{x}_m$, as well as all measurements that directly involve them, can be discarded.

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 of interest. We observe that since in the case of motion tracking under consideration (i) no loop closures occur, (ii) the number of landmarks seen by the robot at any time instant is limited, and (iii) each feature is not observed over very long time periods, $\mathbf{A}$ 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.

3 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 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 2.2, 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

$$\mathbf{A}(k') = \mathbf{\Lambda}_k^p + \mathbf{\Lambda}_{\mathcal{S}_a(k')}^h\big(\hat{\mathbf{x}}_r(k'), \hat{\mathbf{x}}_n(k')\big) + \mathbf{\Lambda}_{m:k'}^f\big(\hat{\mathbf{x}}_r(k'), \hat{\mathbf{x}}_n(k')\big) \tag{30}$$

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:

$$\begin{aligned}\mathbf{A}^{\text{mar}}_{\text{full}}(k') &= \mathbf{\Lambda}^p_0 + \mathbf{\Lambda}^h_{\mathcal{S}_m}(\hat{\mathbf{x}}_m(k), \hat{\mathbf{x}}_r(k)) + \mathbf{\Lambda}^f_{0:m}(\hat{\mathbf{x}}_m(k), \hat{\mathbf{x}}_r(k)) + \mathbf{\Lambda}^h_{\mathcal{S}_a(k')}(\hat{\mathbf{x}}_r(k'), \hat{\mathbf{x}}_n(k')) + \mathbf{\Lambda}^f_{m:k'}(\hat{\mathbf{x}}_r(k'), \hat{\mathbf{x}}_n(k')) \\ &= \begin{bmatrix} \mathbf{A}_{mm}(k) & \mathbf{A}_{mr}(k)\mathbf{\Pi}_r \\ \mathbf{\Pi}_r^T\mathbf{A}_{rm}(k) & \mathbf{\Pi}_r^T\mathbf{A}_{rr}(k)\mathbf{\Pi}_r \end{bmatrix} + \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{A}(k') - \mathbf{\Lambda}^p_k \end{bmatrix}\end{aligned} \tag{31}$$

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 $\mathbf{\Lambda}^h_{\mathcal{S}_m}(\hat{\mathbf{x}}_m(k), \hat{\mathbf{x}}_r(k))$ and $\mathbf{\Lambda}^h_{\mathcal{S}_a(k')}(\hat{\mathbf{x}}_r(k'), \hat{\mathbf{x}}_n(k'))$, taken together, represent all the available measurement information. Similarly, $\mathbf{\Lambda}^f_{0:m}(\hat{\mathbf{x}}_m(k), \hat{\mathbf{x}}_r(k))$ and $\mathbf{\Lambda}^f_{m:k'}(\hat{\mathbf{x}}_r(k'), \hat{\mathbf{x}}_n(k'))$ together represent all the available process-model information, while $\mathbf{\Lambda}^p_0$ 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

$$\mathbf{A}^{\text{mar}}_{\text{full}}(k') = \mathbf{\Lambda}^h_{\mathcal{S}_m}(\hat{\mathbf{x}}_m(k), \hat{\mathbf{x}}_r(k)) + \mathbf{\Lambda}^f_{0:m}(\hat{\mathbf{x}}_m(k), \hat{\mathbf{x}}_r(k)) + \mathbf{\Lambda}^h_{\mathcal{S}_a(k')}(\hat{\mathbf{x}}_r(k'), \hat{\mathbf{x}}_n(k')) + \mathbf{\Lambda}^f_{m:k'}(\hat{\mathbf{x}}_r(k'), \hat{\mathbf{x}}_n(k')) \tag{32}$$

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 equal to:

$$\mathbf{A}^{\text{nm}}_{\text{full}}(k') = \mathbf{\Lambda}^h_{\mathcal{S}}(\hat{\mathbf{x}}_m(k'), \hat{\mathbf{x}}_r(k'), \hat{\mathbf{x}}_n(k')) + \mathbf{\Lambda}^f_{0:k'}(\hat{\mathbf{x}}_m(k'), \hat{\mathbf{x}}_r(k'), \hat{\mathbf{x}}_n(k')) \tag{33}$$

where $\mathcal{S} = \mathcal{S}_m \bigcup \mathcal{S}_a(k')$ be the set of all measurements recorded in the time interval $[0, k']$.

A key result that we prove is that

$$\text{rank}(\mathbf{A}^{\text{mar}}_{\text{full}}(k')) > \text{rank}(\mathbf{A}^{\text{nm}}_{\text{full}}(k')) \tag{34}$$

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. In the following subsections, we will provide the proof for the above result by analyzing the rank of matrices $\mathbf{A}^{\text{nm}}_{\text{full}}$ and $\mathbf{A}^{\text{mar}}_{\text{full}}$ for the case of both 2D and 3D motion, with common measurement methods. Please refer to Table 1 for definitions of symbols used in the derivation.

3.1 Structure of the matrices $\mathbf{A}^{\text{nm}}_{\text{full}}(k')$ and $\mathbf{A}^{\text{mar}}_{\text{full}}(k')$

We start by examining the structure of matrices $\mathbf{A}^{\text{nm}}_{\text{full}}(k')$ and $\mathbf{A}^{\text{mar}}_{\text{full}}(k')$. Using the definition of these matrices in (32) and (33), we see that we can rewrite these matrices as follows:

$$\mathbf{A}^{\text{nm}}_{\text{full}}(k') = \sum_{(i,j)\in\mathcal{S}} \mathbf{H}^T_{ij}(k')\mathbf{R}^{-1}_{ij}\mathbf{H}_{ij}(k') + \sum_{i=0}^{k'-1} \mathbf{G}^T_i(k')\mathbf{Q}^{-1}_i\mathbf{G}_i(k') \tag{35}$$

$$= \underbrace{\begin{bmatrix} \mathbf{G}_0(k') \\ \vdots \\ \mathbf{G}_{k'-1}(k') \\ \hline \vdots \\ \mathbf{H}_{ij}(k') \\ \vdots \end{bmatrix}^T}_{\mathbf{W}^T(k')} \underbrace{\left[\begin{array}{ccc|ccc} \mathbf{Q}^{-1}_0 & \dots & \mathbf{0} & \dots & \dots & \mathbf{0} \\ \mathbf{0} & \ddots & \mathbf{0} & \dots & \dots & \mathbf{0} \\ \mathbf{0} & \ddots & \mathbf{Q}^{-1}_{k'-1} & \mathbf{0} & \dots & \mathbf{0} \\ \hline \mathbf{0} & \dots & \mathbf{0} & \ddots & \ddots & \mathbf{0} \\ \mathbf{0} & \dots & \dots & \mathbf{0} & \mathbf{R}^{-1}_{ij} & \mathbf{0} \\ \mathbf{0} & \dots & \dots & \dots & \mathbf{0} & \ddots \end{array}\right]}_{\mathbf{S}} \underbrace{\begin{bmatrix} \mathbf{G}_0(k') \\ \vdots \\ \mathbf{G}_{k'-1}(k') \\ \hline \vdots \\ \mathbf{H}_{ij}(k') \\ \vdots \end{bmatrix}}_{\mathbf{W}(k')} \tag{36}$$

$$= \mathbf{W}^T(k')\mathbf{S}\mathbf{W}(k') \tag{37}$$

and

$$\mathbf{A}_{\text{full}}^{\text{mar}}(k') = \underbrace{\left[\begin{array}{c} \mathbf{G}_0(k) \\ \vdots \\ \hdashline \mathbf{G}_{m-1}(k) \\ \mathbf{G}_m(k') \\ \vdots \\ \mathbf{G}_{k'-1}(k') \\ \hline \vdots \\ \mathbf{H}_{ij}(k) \\ \vdots \\ \hdashline \vdots \\ \mathbf{H}_{ij}(k') \\ \vdots \end{array}\right]^T}_{\mathbf{W}^T(k,k')} \underbrace{\left[\begin{array}{ccc|ccc} \mathbf{Q}_0^{-1} & \dots & \mathbf{0} & \dots & \dots & \mathbf{0} \\ \mathbf{0} & \ddots & \mathbf{0} & \dots & \dots & \mathbf{0} \\ \mathbf{0} & \ddots & \mathbf{Q}_{k'-1}^{-1} & \mathbf{0} & \dots & \mathbf{0} \\ \hline \mathbf{0} & \dots & \mathbf{0} & \ddots & \ddots & \mathbf{0} \\ \mathbf{0} & \dots & \dots & \mathbf{0} & \mathbf{R}_{ij}^{-1} & \mathbf{0} \\ \mathbf{0} & \dots & \dots & \dots & \mathbf{0} & \ddots \end{array}\right]}_{\mathbf{S}} \underbrace{\left[\begin{array}{c} \mathbf{G}_0(k) \\ \vdots \\ \hdashline \mathbf{G}_{m-1}(k) \\ \mathbf{G}_m(k') \\ \vdots \\ \mathbf{G}_{k'-1}(k') \\ \hline \vdots \\ \mathbf{H}_{ij}(k) \\ \vdots \\ \hdashline \vdots \\ \mathbf{H}_{ij}(k') \\ \vdots \end{array}\right]}_{\mathbf{W}(k,k')} \tag{38}$$

$$= \mathbf{W}^T(k,k')\ \mathbf{S}\ \mathbf{W}(k,k') \tag{39}$$

The first k' block rows of the matrices $\mathbf{W}(k')$ and $\mathbf{W}(k,k')$ correspond to the odometry measurements, while the remaining block rows correspond to the robot-to-landmark measurements ($\mathbf{H}_{ij}$ and $\mathbf{G}_i$ are defined in Eq. (12) and (13), respectively). Therefore, these matrices have $k'd_r + ld_m$ rows and $(k'+1)d_r + nd_\ell$ columns each. We once again stress that $\mathbf{A}_{\text{full}}^{\text{nm}}(k')$ and $\mathbf{A}_{\text{full}}^{\text{mar}}(k')$ have the same structure, and the only difference between these matrices is the state estimates at which some of the Jacobians are evaluated. As a result, the matrices $\mathbf{W}(k')$ and $\mathbf{W}(k,k')$ also have the same structure, and the only difference lies in the state estimates at which the Jacobians are evaluated. Specifically, we here consider the case where at time step k the m oldest robot poses, $\mathbf{r}_{0:m-1}$, are marginalized along with the m_l oldest landmarks, which are seen *only* by marginalized poses. As a result, in the matrix $\mathbf{W}(k,k')$ the Jacobians of all measurements that that involve marginalized poses (the matrices $\mathbf{G}_i$ for $i = 0, \dots m-1$, and the matrices $\mathbf{H}_{ij}$ where $(i,j) \in \mathcal{S}_m$) are evaluated using the state estimates available at time-step k, while all other Jacobians are evaluated using the estimates available at time-step k'. The parts of the matrix $\mathbf{W}(k,k')$ that are evaluated using different state estimates are indicated with the dashed lines in (38). On the other hand, in the matrix $\mathbf{W}(k')$, *all* Jacobians are evaluated using the state estimates at time-step k'.

In the following derivations, for clarity, we will employ the following ordering of the variables: all the robot poses will come first in the temporal order, followed by all the landmarks. As a result, using the definition of the matrices $\mathbf{H}_{ij}$ and $\mathbf{G}_i$ in (12) and (13), respectively, we obtain the structure of the matrices $\mathbf{W}(k')$ and $\mathbf{W}(k,k')$ is as follows:

$$\mathbf{W} = \left[\begin{array}{c} \mathbf{G}_0 \\ \vdots \\ \mathbf{G}_{k'-1} \\ \hline \vdots \\ \mathbf{H}_{ij} \\ \vdots \end{array}\right] = \left[\begin{array}{ccc|c} -\mathbf{\Phi}_{R_0} & \mathbf{I}_{d_r} & \dots & \mathbf{0}_{d_r \times d_r} & \mathbf{0}_{d_r \times nd_\ell} \\ \vdots & \ddots & \ddots & \vdots & \vdots \\ \mathbf{0}_{d_r \times d_r} & \dots & -\mathbf{\Phi}_{R_{k'-1}} & \mathbf{I}_{d_r} & \mathbf{0}_{d_r \times nd_\ell} \\ \hline \mathbf{H}_{R_0} & \dots & \dots & \mathbf{0}_{d_m l_0 \times d_r} & \mathbf{H}_{L_0} \\ \mathbf{0}_{d_m l_i \times d_r} & \mathbf{H}_{R_1} & \dots & \vdots & \vdots \\ \vdots & \dots & \ddots & \mathbf{0}_{d_m l_{i+1} \times d_r} & \vdots \\ \mathbf{0}_{d_m l_{k'} \times d_r} & \dots & \dots & \mathbf{H}_{R_{k'}} & \mathbf{H}_{L_{k'}} \end{array}\right] \tag{40}$$

where the time-step indices have been omitted, since the above structure is shared by both $\mathbf{W}(k')$ and $\mathbf{W}(k,k')$. In the above, the matrices $\mathbf{H}_{R_i}$ and $\mathbf{H}_{L_i}$ are block matrices that contain the Jacobians of all the robot-to-landmark measurements at time-step i. Specifically, if at time-step i the robot observes l_i landmarks then $\mathbf{H}_{R_i}$ is an $l_i \times 1$ block vector, containing the Jacobians $\mathbf{H}_{R_{ij}}$, while $\mathbf{H}_{L_i}$ contains l_i block rows, where each row contains the Jacobian $\mathbf{H}_{L_{ij}}$ at the jth position:

$$\mathbf{H}_{R_i} = \begin{bmatrix} \mathbf{H}_{R_{ij_1}} \\ \mathbf{H}_{R_{ij_2}} \\ \vdots \\ \mathbf{H}_{R_{ij_{l_i}}} \end{bmatrix}, \quad \mathbf{H}_{L_i} = \begin{bmatrix} \mathbf{0} & \mathbf{H}_{L_{ij_1}} & \dots & \dots & \mathbf{0} \\ \mathbf{H}_{L_{ij_2}} & \mathbf{0} & \dots & \dots & \mathbf{0} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{0} & \dots & \mathbf{H}_{L_{ij_{l_i}}} & \dots & \mathbf{0} \end{bmatrix} \tag{41}$$

Exploiting the special structure shown above allows us to prove the following result:

Lemma 1. *The rank of the matrix* $\mathbf{A}_{\text{full}}^{\text{nm}}(k')$ *is given by:*

$$\text{rank}(\mathbf{A}_{\text{full}}^{\text{nm}}(k')) = k'd_r + \text{rank}(\mathbf{M}(k')) \tag{42}$$

where

$$\mathbf{M}(k') = [\mathbf{M}_R(k') \;\; \mathbf{M}_L(k')] \tag{43}$$

with

$$\mathbf{M}_R(k') = \begin{bmatrix} \mathbf{H}_{R_0}(k') \\ \mathbf{H}_{R_1}(k')\mathbf{\Phi}_{R_0}(k') \\ \vdots \\ \mathbf{H}_{R_{k'-1}}(k')\mathbf{\Phi}_{R_{k'-2}}(k')\ldots\mathbf{\Phi}_{R_0}(k') \\ \mathbf{H}_{R_{k'}}(k')\mathbf{\Phi}_{R_{k'-1}}(k')\ldots\mathbf{\Phi}_{R_0}(k') \end{bmatrix}, \;\; \mathbf{M}_L(k') = \begin{bmatrix} \mathbf{H}_{L_0}(k') \\ \vdots \\ \\ \vdots \\ \mathbf{H}_{L_{k'}}(k') \end{bmatrix} \tag{44}$$

Similarly, the rank of the matrix $\mathbf{A}_{\text{full}}^{\text{mar}}(k')$ *is given by:*

$$\text{rank}(\mathbf{A}_{\text{full}}^{\text{mar}}(k')) = k'd_r + \text{rank}(\mathbf{M}(k,k')) \tag{45}$$

where

$$\mathbf{M}(k,k') = [\mathbf{M}_R(k,k') \;\; \mathbf{M}_L(k,k')] \tag{46}$$

with

$$\mathbf{M}_R(k,k') = \begin{bmatrix} \mathbf{H}_{R_0}(k) \\ \mathbf{H}_{R_1}(k)\mathbf{\Phi}_{R_0}(k) \\ \vdots \\ \mathbf{H}_{R_{m-1}}(k)\mathbf{\Phi}_{R_{m-2}}(k)\ldots\mathbf{\Phi}_{R_0}(k) \\ \mathbf{H}_{R_m}(k')\mathbf{\Phi}_{R_{m-1}}(k)\ldots\mathbf{\Phi}_{R_0}(k) \\ \mathbf{H}_{R_{m+1}}(k')\mathbf{\Phi}_{R_m}(k')\ldots\mathbf{\Phi}_{R_0}(k) \\ \vdots \\ \mathbf{H}_{R_{k'-1}}(k')\mathbf{\Phi}_{R_{k'-2}}(k')\ldots\mathbf{\Phi}_{R_0}(k) \\ \mathbf{H}_{R_{k'}}(k')\mathbf{\Phi}_{R_{k-1}}(k')\ldots\mathbf{\Phi}_{R_0}(k) \end{bmatrix}, \;\; \mathbf{M}_L(k,k') = \begin{bmatrix} \mathbf{H}_{L_0}(k) \\ \vdots \\ \mathbf{H}_{L_{m-1}}(k) \\ \mathbf{H}_{L_m}(k') \\ \vdots \\ \mathbf{H}_{L_{k'}}(k') \end{bmatrix} \tag{47}$$

Proof. See Appendix A. □

Note that the above result does not make any assumptions about the dimension of the robot or landmark poses or about the type of measurements available. In the following subsections, we analyze the rank of the matrices $\mathbf{A}_{\text{full}}^{\text{nm}}(k')$ and $\mathbf{A}_{\text{full}}^{\text{mar}}(k')$ for 2D and 3D motion, with commonly used measurement models.

3.2 Robot motion in 2D

3.2.1 Description of the motion and measurement models

For the case of 2D motion, a robot pose is characterized by its 2D position and orientation, ie., $\mathbf{r}_i = [\mathbf{p}_{R_i}^T \;\; \phi_{R_i}]$ and $d_r = 3$, while a landmark is described by its 2D position, ie., $\mathbf{l}_j = \mathbf{p}_{L_j}$ and $d_\ell = 2$. In the propagation step, the pose change between two consecutive time steps is estimated based on odometry measurements. The propagation equations are given by:

$$\hat{\mathbf{p}}_{R_{i+1}} = \hat{\mathbf{p}}_{R_i} + \mathbf{C}(\hat{\phi}_{R_i})^{R_i}\hat{\mathbf{p}}_{R_{i+1}} \tag{48}$$

$$\hat{\phi}_{R_{i+1}} = \hat{\phi}_{R_i} + {}^{R_i}\hat{\phi}_{R_{i+1}} \tag{49}$$

where $\mathbf{C}(.)$ denotes the 2×2 rotation matrix, and ${}^{R_i}\hat{\mathbf{x}}_{R_{i+1}} = [{}^{R_i}\hat{\mathbf{p}}_{R_{i+1}} \;\; {}^{R_i}\hat{\phi}_{R_{i+1}}]^T$ is the estimate of the change in robot pose between time steps i and $i+1$, based on odometry measurements. The estimate is corrupted by zero-mean, white Gaussian noise $\mathbf{w}_i$, with covariance matrix $\mathbf{Q}_i$. The process model is nonlinear, and can be described by the following nonlinear function:

$$\mathbf{x}_{i+1} = \mathbf{f}(\mathbf{x}_i, {}^{R_i}\hat{\mathbf{x}}_{R_{i+1}} + \mathbf{w}_i) \tag{50}$$

The linearized error-state propagation equation for the above state propagation equation is given by:

$$\tilde{\mathbf{x}}_{R_{i+1}} = \mathbf{\Phi}_{R_i}\tilde{\mathbf{x}}_{R_i} + \mathbf{\Gamma}_{R_i}\mathbf{w}_i \tag{51}$$

where $\mathbf{\Phi}_{R_i}$ and $\mathbf{\Gamma}_{R_i}$ are obtained from the state propagation equations (48):

$$\mathbf{\Phi}_{R_i} = \begin{bmatrix} \mathbf{I}_2 & \mathbf{J}\mathbf{C}(\hat{\phi}_{R_i})^{R_i}\hat{\mathbf{p}}_{R_{i+1}} \\ \mathbf{0}_{1\times 2} & 1 \end{bmatrix} \tag{52}$$

$$\equiv \begin{bmatrix} \mathbf{I}_2 & \mathbf{J}(\hat{\mathbf{p}}_{R_{i+1}} - \hat{\mathbf{p}}_{R_i}) \\ \mathbf{0}_{1\times 2} & 1 \end{bmatrix} \tag{53}$$

$$\mathbf{\Gamma}_{R_i} = \begin{bmatrix} \mathbf{C}(\hat{\phi}_{R_i}) & \mathbf{0}_{2\times 1} \\ \mathbf{0}_{1\times 2} & 1 \end{bmatrix} \tag{54}$$

with $\mathbf{J} \triangleq \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$. The form of the propagation equations presented above is general, and holds for any robot kinematic model (e.g., unicycle, bicycle).

The robot-to-landmark measurement $\mathbf{z}_i ij$ is a function of the relative position of the landmark with respect to the robot, which we denote by $\Delta\mathbf{p}^{ij} = \mathbf{C}^T(\phi_{R_i})(\mathbf{p}_{L_j} - \mathbf{p}_{R_i})$. Therefore,

$$\mathbf{z}_{ij} = \mathbf{h}(\Delta\mathbf{p}^{ij}) + \mathbf{n}_{ij} = \mathbf{h}(\mathbf{r}_i, \mathbf{l}_j) + \mathbf{n}_{ij} \tag{55}$$

The measurement function $\mathbf{h}'$ can have several forms. For example, it can be the direct measurement of relative position, a pair of range and bearing measurements, bearing-only measurements from a monocular camera, etc. Generally, the measurement function is nonlinear. Using the chain rule of differentiation, we can compute the Jacobians with respect to the robot pose and the landmark position, as:

$$\mathbf{H}_{R_{ij}} = \frac{\partial\mathbf{z}_{ij}}{\partial\Delta\mathbf{p}^{ij}}\frac{\partial\Delta\mathbf{p}^{ij}}{\partial\mathbf{r}_i} = \nabla\mathbf{h}_{ij}\mathbf{C}^T(\hat{\phi}_{R_i})[-\mathbf{I}_2 \quad -\mathbf{J}(\hat{\mathbf{p}}_{L_j} - \hat{\mathbf{p}}_{R_i})] \tag{56}$$

$$\mathbf{H}_{L_{ij}} = \frac{\partial\mathbf{z}_{ij}}{\partial\Delta\mathbf{p}^{ij}}\frac{\partial\Delta\mathbf{p}^{ij}}{\partial\mathbf{l}_j} = \nabla\mathbf{h}_{ij}\mathbf{C}^T(\hat{\phi}_{R_i}) \tag{57}$$

where $\nabla\mathbf{h}_{ij}$ is the Jacobian of $\mathbf{h}$ with respect to the relative robot-landmark position $\Delta\mathbf{p}^{ij}$. From Eq. (56), we note that:

$$\mathbf{H}_{R_{ij}} = \mathbf{H}_{L_{ij}}[-\mathbf{I}_2 \quad -\mathbf{J}(\hat{\mathbf{p}}_{L_j} - \hat{\mathbf{p}}_{R_i})] \tag{58}$$

$$= \mathbf{H}_{L_{ij}}\mathbf{H}'_{R_{ij}} \tag{59}$$

with $\mathbf{H}'_{R_{ij}} = [-\mathbf{I}_2 \quad -\mathbf{J}(\hat{\mathbf{p}}_{L_j} - \hat{\mathbf{p}}_{R_i})]$. This is a useful property, which we will use in the analysis that follows.

3.2.2 Rank of $\mathbf{A}^{\text{nm}}_{\text{full}}(k')$

We will first employ the result of Lemma 1 to compute the rank of the matrix $\mathbf{A}^{\text{nm}}_{\text{full}}(k')$ in the case of 2D motion. Using the result of (58), we can obtain the following decomposition of the matrix $\mathbf{M}(k')$:

$$\mathbf{M}(k') = \mathbf{D}(k')\mathbf{K}(k') \tag{60}$$

where $\mathbf{D}(k')$ is a block diagonal matrix:

$$\mathbf{D}(k') = \mathbf{Diag}\left(\mathbf{H}_{L_{ij}}(k')\right), (i,j) \in \mathcal{S} \tag{61}$$

and $\mathbf{K}(k')$ is given by

$$\mathbf{K}(k') = [\mathbf{K}_R(k') \;\; \mathbf{K}_L] \tag{62}$$

with

$$\mathbf{K}_R(k') = \begin{bmatrix} \mathbf{H}'_{R_0}(k') \\ \mathbf{H}'_{R_1}(k')\mathbf{\Phi}_{R_0}(k') \\ \vdots \\ \mathbf{H}'_{R_{k'-1}}(k')\mathbf{\Phi}_{R_{k'-2}}(k')\ldots\mathbf{\Phi}_{R_0}(k') \\ \mathbf{H}'_{R_{k'}}(k')\mathbf{\Phi}_{R_{k'-1}}(k')\ldots\mathbf{\Phi}_{R_0}(k') \end{bmatrix}, \;\; \mathbf{K}_L = \begin{bmatrix} \mathbf{H}'_{L_0} \\ \mathbf{H}'_{L_1} \\ \vdots \\ \mathbf{H}'_{L_{k'-1}} \\ \mathbf{H}'_{L_{k'}} \end{bmatrix} \tag{63}$$

In the above, $\mathbf{H}'_{R_i}(k')$ is the matrix that results from factoring out $\mathbf{H}_{L_{ij}}(k')$ from each row of $\mathbf{H}_{R_i}(k')$, ie.,

$$\mathbf{H}'_{R_i}(k') = \begin{bmatrix} \mathbf{H}'_{R_{ij_1}}(k') \\ \mathbf{H}'_{R_{ij_2}}(k') \\ \vdots \\ \mathbf{H}'_{R_{ij_{l_i}}}(k') \end{bmatrix} = \begin{bmatrix} -\mathbf{I}_2 & -\mathbf{J}(\hat{\mathbf{p}}_{L_{j_1}}(k') - \hat{\mathbf{p}}_{R_i}(k')) \\ -\mathbf{I}_2 & -\mathbf{J}(\hat{\mathbf{p}}_{L_{j_2}}(k') - \hat{\mathbf{p}}_{R_i}(k')) \\ \vdots & \\ -\mathbf{I}_2 & -\mathbf{J}(\hat{\mathbf{p}}_{L_{j_{l_i}}}(k') - \hat{\mathbf{p}}_{R_i}(k')) \end{bmatrix}, \quad \mathbf{H}'_{L_i} = \begin{bmatrix} \mathbf{0} & \mathbf{I}_2 & \dots & \dots & \mathbf{0} \\ \mathbf{I}_2 & \mathbf{0} & \dots & \dots & \mathbf{0} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{0} & \dots & \mathbf{I}_2 & \dots & \mathbf{0} \end{bmatrix} \tag{64}$$

Our goal is to find the rank of the matrix $\mathbf{M}(k')$, which will in turn allow us to compute the rank of $\mathbf{A}^{\text{nm}}_{\text{full}}(k')$ using the result of Lemma 1. To this end, we will use the following result [19, 4.5.1]:

$$\begin{aligned} \text{rank}(\mathbf{M}(k')) &= \text{rank}(\mathbf{D}(k')\mathbf{K}(k')) \\ &= \text{rank}(\mathbf{K}(k')) - \dim(\mathcal{N}(\mathbf{D}(k')) \bigcap \mathcal{R}(\mathbf{K}(k'))) \end{aligned} \tag{65}$$

where $\mathcal{N}$ and $\mathcal{R}$ denote the null space and the range of a matrix, respectively. The following two lemmas allow us to obtain the final result:

Lemma 2. *When the robot is moving in 2D, and observes n landmarks in total,*

$$\text{rank}(\mathbf{K}(k')) = 2n \tag{66}$$

Proof. See Appendix B.1. □

Lemma 3. *When the robot-to-landmark measurements are relative positions, relative ranges, or relative bearings, and for general robot motion,*

$$\dim(\mathcal{N}(\mathbf{D}(k')) \bigcap \mathcal{R}(\mathbf{K}(k'))) = 0 \tag{67}$$

Proof. See Appendix B.2. □

Combining the results of (42), (65), (66) and (67), we conclude that when the robot is moving in 2D, and observes landmarks using relative-position, relative-range, or relative-bearing sensors, then

$$\text{rank}(\mathbf{A}^{\text{nm}}_{\text{full}}(k')) = 3k' + 2n \tag{68}$$

It is important to note that the dimension of the matrix $\mathbf{A}^{\text{nm}}_{\text{full}}(k')$ is equal to $3(k'+1)+2n$ (since $\mathbf{A}^{\text{nm}}_{\text{full}}(k')$ is the information matrix for the entire state, which contains $k'+1$ robot poses and n landmarks). Therefore, the matrix is rank deficient by 3, and has a nullspace of dimension 3. By inspection, a basis for the nullspace can be found, and is given by the columns of the matrix:

$$\mathbf{N}(k') = \begin{bmatrix} \mathbf{I}_2 & \mathbf{J}\hat{\mathbf{p}}_{R_0}(k') \\ \mathbf{0}_{1\times 2} & 1 \\ \vdots & \vdots \\ \mathbf{I}_2 & \mathbf{J}\hat{\mathbf{p}}_{R_{k'}}(k') \\ \mathbf{0}_{1\times 2} & 1 \\ \mathbf{I}_2 & \mathbf{J}\hat{\mathbf{p}}_{L_1}(k') \\ \vdots & \vdots \\ \mathbf{I}_2 & \mathbf{J}\hat{\mathbf{p}}_{L_n}(k') \end{bmatrix} \tag{69}$$

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).

3.2.3 Rank of $\mathbf{A}_{\text{full}}^{\text{mar}}(k')$

To compute the rank of the matrix $\mathbf{A}_{\text{full}}^{\text{mar}}(k')$, we proceed in a manner similar to the analysis for the rank of $\mathbf{A}_{\text{full}}^{\text{nm}}(k')$. We first employ the property in (58) to perform the following decomposition:

$$\mathbf{M}(k,k') = \mathbf{D}(k,k')\mathbf{K}(k,k') \tag{70}$$

where $\mathbf{D}(k,k')$ is the block diagonal matrix:

$$\mathbf{D}(k,k') = \begin{bmatrix} \mathbf{D}_m(k) & \mathbf{0} \\ \mathbf{0} & \mathbf{D}_a(k') \end{bmatrix}, \quad \text{with} \tag{71}$$

$$\mathbf{D}_m(k) = \mathbf{Diag}(\mathbf{H}_{L_{ij}}(k)), \qquad (i,j) \in \mathcal{S}_m \tag{72}$$

$$\mathbf{D}_a(k') = \mathbf{Diag}(\mathbf{H}_{L_{ij}}(k')), \qquad (i,j) \in \mathcal{S}_a \tag{73}$$

and

$$\mathbf{K}(k,k') = [\mathbf{K}_R(k,k') \;\; \mathbf{K}_L] \tag{74}$$

with

$$\mathbf{K}_R(k,k') = \begin{bmatrix} \mathbf{H}'_{R_0}(k) \\ \mathbf{H}'_{R_1}(k)\mathbf{\Phi}_{R_0}(k) \\ \vdots \\ \mathbf{H}'_{R_{m-1}}(k)\mathbf{\Phi}_{R_{m-2}}(k)\dots\mathbf{\Phi}_{R_0}(k) \\ \mathbf{H}'_{R_m}(k')\mathbf{\Phi}_{R_{m-1}}(k)\dots\mathbf{\Phi}_{R_0}(k) \\ \mathbf{H}'_{R_{m+1}}(k')\mathbf{\Phi}_{R_m}(k')\dots\mathbf{\Phi}_{R_0}(k) \\ \vdots \\ \mathbf{H}'_{R_{k'-1}}(k')\mathbf{\Phi}_{R_{k'-2}}(k')\dots\mathbf{\Phi}_{R_0}(k) \\ \mathbf{H}'_{R_{k'}}(k')\mathbf{\Phi}_{R_{k-1}}(k')\dots\mathbf{\Phi}_{R_0}(k) \end{bmatrix}, \;\; \mathbf{K}_L = \begin{bmatrix} \mathbf{H}'_{L_0} \\ \vdots \\ \mathbf{H}'_{L_{m-1}} \\ \mathbf{H}'_{L_m} \\ \vdots \\ \mathbf{H}'_{L_{k'}} \end{bmatrix} \tag{75}$$

where the above terms are defined similarly to those in (63) and (64). Again, the difference lies in the state estimates used to evaluate those Jacobians that involve marginalized states. It is interesting to note that the matrix $\mathbf{K}_L$ above is the same as that in (64).

To compute the rank of the matrix $\mathbf{M}(k')$ we will once again use the property:

$$\text{rank}(\mathbf{M}(k,k')) = \text{rank}(\mathbf{D}(k,k')\mathbf{K}(k,k')) = \text{rank}(\mathbf{K}(k,k')) - \dim(\mathcal{N}(\mathbf{D}(k,k')) \bigcap \mathcal{R}(\mathbf{K}(k,k'))) \tag{76}$$

Similarly to the previous section, we evaluate the two terms in the above equation separately, and state the results in the following two lemmas:

Lemma 4. *When the robot is moving in 2D, and observes n landmarks in total,*

$$\text{rank}(\mathbf{K}(k,k')) = 2n+1 \tag{77}$$

Proof. See Appendix B.3. □

Lemma 5. *When the robot-to-landmark measurements are relative positions, relative ranges, or relative bearings, and for general robot motion,*

$$\dim(\mathcal{N}(\mathbf{D}(k,k')) \bigcap \mathcal{R}(\mathbf{K}(k,k'))) = 0 \tag{78}$$

Proof. See Appendix B.4. □

Using the above results, in conjunction with (45), we obtain the following key relationship:

$$\text{rank}(\mathbf{A}_{\text{full}}^{\text{mar}}(k')) = 3k' + 2n + 1 = \text{rank}(\mathbf{A}_{\text{full}}^{\text{nm}}(k')) + 1 \tag{79}$$

We have therefore proved that the rank of the information matrix is increased by one, compared to the case where all the Jacobians are evaluated using the same state estimates. Moreover, a basis for the nullspace of the matrix $\mathbf{A}_{\text{full}}^{\text{mar}}(k')$ (which is now of dimension 2) can be found by inspection, and is given by the first two columns of the matrix $\mathbf{N}(k')$ in (69). We thus see that the "missing" direction of the nullspace is the one that corresponds to the global orientation of the state vector. In turn, this implies that the estimator erroneously believes the global orientation to be observable, based on the available measurements. The ramifications of this result are further discussed in Section 3.4.

3.3 Robot motion in 3D

3.3.1 Description of the motion and measurement models

In the case of 3D motion, typically either an inertial measurement unit (IMU) or a statistical motion model (e.g., constant velocity, constant acceleration) is used for propagating the state estimates. We here focus on the case where an IMU is used for 3D motion tracking. For simplicity, in this analysis, we assume that the IMU biases are known *a priori*, and don't need to be estimated online. Therefore the IMU state vector comprises the IMU orientation, position, and velocity. We employ a unit-quaternion description of orientation, and therefor the IMU state is given by:

$$\mathbf{r} = \begin{bmatrix} \mathbf{q}_R \\ \mathbf{p}_R \\ \mathbf{v}_R \end{bmatrix} \tag{80}$$

Where $\mathbf{q}_R$ is the unit quaternion representing the rotation between the robot (IMU) frame and the global frame, while $\mathbf{p}_R$ and $\mathbf{v}_R$ represent the IMU position and velocity in the global frame, respectively. The discrete-time state propagation equations are computed by integration of the nonlinear continuous-time system model [20], and by linearization we obtain the linearized error-state propagation equations:

$$\delta\mathbf{r}_{i+1} = \mathbf{\Phi}_{R_i}\delta\mathbf{r}_i + \mathbf{w}_i \tag{81}$$

where $\delta\mathbf{r}_i$ is the 9×1 error-state vector, defined as:

$$\delta\mathbf{r}_i = \begin{bmatrix} \delta\boldsymbol{\theta}_{R_i} \\ \tilde{\mathbf{p}}_{R_i} \\ \tilde{\mathbf{v}}_{R_i} \end{bmatrix}$$

In the above expression, $\tilde{\mathbf{p}}_{R_i}$ and $\tilde{\mathbf{v}}_{R_i}$ are the estimation errors (difference between the actual and estimated value) for the IMU position and velocity, while $\delta\boldsymbol{\theta}_{R_i}$ is the 3×1 orientation error vector [20]. The state-transition matrix $\mathbf{\Phi}_{R_i}$ in (81) is given by:

$$\mathbf{\Phi}_{R_i} = \begin{bmatrix} {}^{R_{i+1}}_{R_i}\mathbf{R} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ -\lfloor(\hat{\mathbf{p}}_{R_{i+1}} - \hat{\mathbf{p}}_{R_i} - \hat{\mathbf{v}}_{R_i}({}^{i+1}_{i}\Delta t) - \frac{\mathbf{g}}{2}({}^{i+1}_{i}\Delta t)^2)\times\rfloor\mathbf{C}^T(\hat{\mathbf{q}}_{R_i}) & \mathbf{I}_3 & {}^{i+1}_{i}\Delta t\mathbf{I}_3 \\ -\lfloor(\hat{\mathbf{v}}_{R_{i+1}} - \hat{\mathbf{v}}_{R_i} - \mathbf{g}({}^{i+1}_{i}\Delta t))\times\rfloor\mathbf{C}^T(\hat{\mathbf{q}}_{R_i}) & \mathbf{0}_{3\times3} & \mathbf{I}_3 \end{bmatrix} \tag{82}$$

where ${}^{i+1}_{i}\Delta t = t_{i+1} - t_i$ is the time difference between time-steps i and $i+1$, $\mathbf{g}$ is the gravitational acceleration vector, and $\mathbf{C}(\hat{\mathbf{q}}_{R_i})$ is the 3×3 rotation matrix corresponding to $\hat{\mathbf{q}}_{R_i}$.

The sensor measurement is a function of the relative position of the landmark with respect to the robot:

$$\mathbf{z}_{ij} = \mathbf{h}(\Delta\mathbf{p}^{ij}) + \mathbf{n}_{ij} = \mathbf{h}(\mathbf{p}_{R_i}, \mathbf{p}_{L_j}) + \mathbf{n}_{ij} \tag{83}$$

where

$$\Delta\mathbf{p}^{ij} = \mathbf{C}(\mathbf{q}_{R_i})(\mathbf{p}_{L_j} - \mathbf{p}_{R_i}) \tag{84}$$

The Jacobians of the measurement model with respect to the robot and landmark states are given by:

$$\begin{aligned} \mathbf{H}_{R_{ij}} &= \tfrac{\partial\mathbf{z}_{ij}}{\partial\Delta\mathbf{p}^{ij}}\tfrac{\partial\Delta\mathbf{p}^{ij}}{\partial\mathbf{r}_i} &&= \nabla\mathbf{h}_{ij}\,\mathbf{C}(\mathbf{q}_{R_i})\begin{bmatrix} \lfloor(\mathbf{p}_{L_j} - \mathbf{p}_{R_i})\times\rfloor\mathbf{C}^T(\mathbf{q}_{R_i}) & -\mathbf{I}_3 & \mathbf{0}_{3\times3} \end{bmatrix} \\ \mathbf{H}_{L_{ij}} &= \tfrac{\partial\mathbf{z}_{ij}}{\partial\Delta\mathbf{p}^{ij}}\tfrac{\partial\Delta\mathbf{p}^{ij}}{\partial\mathbf{l}_j} &&= \nabla\mathbf{h}_{ij}\,\mathbf{C}(\mathbf{q}_{R_i}) \end{aligned} \tag{85}$$

where $\nabla\mathbf{h}_{ij}$ is the Jacobian of $\mathbf{h}$ with respect to the relative robot-landmark position $\Delta\mathbf{p}^{ij}$. Similarly to Section 3.2, we can write:

$$\begin{aligned} \mathbf{H}_{R_{ij}} &= \mathbf{H}_{L_{ij}}\begin{bmatrix} \lfloor(\mathbf{p}_{L_j} - \mathbf{p}_{R_i})\times\rfloor\mathbf{C}^T(\mathbf{q}_{R_i}) & -\mathbf{I}_3 & \mathbf{0}_{3\times3} \end{bmatrix} \\ &= \mathbf{H}_{L_{ij}}\mathbf{H}'_{R_{ij}} \end{aligned} \tag{86}$$

3.3.2 Rank of $\mathbf{A}_{\text{full}}^{\text{nm}}(k')$

We now employ the result of Lemma 1 to compute the rank of the matrix $\mathbf{A}_{\text{full}}^{\text{nm}}(k')$ in the case of 3D motion with an IMU used for state propagation. In this case, we have $d_r = 9$, and $d_\ell = 3$. Owing to the similar structure of the 2D and 3D measurement models (see (58) and (86)), the analysis is very similar. Specifically, we can obtain a decomposition of the matrix $\mathbf{M}(k')$ that appears in Lemma 1 in the same way as in (60)-(63), with

$$\mathbf{H}'_{R_i}(k') = \begin{bmatrix} \mathbf{H}'_{R_{ij_1}}(k') \\ \mathbf{H}'_{R_{ij_2}}(k') \\ \vdots \\ \mathbf{H}'_{R_{ij_{l_i}}}(k') \end{bmatrix} = \begin{bmatrix} \lfloor (\mathbf{p}_{L_{j_1}}(k') - \mathbf{p}_{R_i}(k')) \times \rfloor \mathbf{C}^T(\mathbf{q}_{R_i}(k')) & -\mathbf{I}_3 & \mathbf{0}_{3\times 3} \\ \lfloor (\mathbf{p}_{L_{j_2}}(k') - \mathbf{p}_{R_i}(k')) \times \rfloor \mathbf{C}^T(\mathbf{q}_{R_i}(k')) & -\mathbf{I}_3 & \mathbf{0}_{3\times 3} \\ \vdots & & \\ \lfloor (\mathbf{p}_{L_{j_{l_i}}}(k') - \mathbf{p}_{R_i}(k')) \times \rfloor \mathbf{C}^T(\mathbf{q}_{R_i}(k')) & -\mathbf{I}_3 & \mathbf{0}_{3\times 3} \end{bmatrix} \tag{87}$$

$$\mathbf{H}'_{L_i} = \begin{bmatrix} \mathbf{0}_{3\times 3} & \mathbf{I}_3 & \dots & \dots & \mathbf{0}_{3\times 3} \\ \mathbf{I}_3 & \mathbf{0}_{3\times 3} & \dots & \dots & \mathbf{0}_{3\times 3} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{0}_{3\times 3} & \dots & \mathbf{I}_3 & \dots & \mathbf{0}_{3\times 3} \end{bmatrix} \tag{88}$$

To compute the rank of the matrix $\mathbf{M}(k')$ we will proceed similarly to Section 3.2, using the result of (65) and the following two lemmas:

Lemma 6. *The rank of the matrix $\mathbf{K}(k')$ for 3D motion tracking with an IMU is $3n + 5$, where n is the number of landmarks.*

Proof. See Appendix C.1. □

Lemma 7. *When the robot-to-landmark measurements are relative-position measurements or camera measurements, and for general motion,*

$$\dim(\mathcal{N}(\mathbf{D}(k')) \bigcap \mathcal{R}(\mathbf{K}(k'))) = 0 \tag{89}$$

Proof. See Appendix C.2. □

Using the above results, we conclude that when the robot-to-landmark measurements are relative-position measurements or camera measurements, in general, we have:

$$\text{rank}(\mathbf{M}(k')) = \underbrace{\text{rank}(\mathbf{K}(k'))}_{3n+5} - \underbrace{\dim(\mathcal{N}(\mathbf{D}(k')) \bigcap \mathcal{R}(\mathbf{K}(k')))}_{0} = 3n + 5 \tag{90}$$

Together with Lemma 1, with $d_r = 9$, we obtain:

$$\text{rank}(\mathbf{A}_{\text{full}}^{\text{nm}}(k')) = 9k' + \underbrace{\text{rank}(\mathbf{M}(k'))}_{3n+5} = 9k' + 3n + 5 \tag{91}$$

Since $\mathbf{A}_{\text{full}}^{\text{nm}}(k')$ is a $(9(k'+1)+3n) \times (9(k'+1)+3n)$ matrix, this result shows that the nullspace of $\mathbf{A}_{\text{full}}^{\text{nm}}(k')$ is of dimension 4. A basis for this nullspace can be found by inspection, and is given by the columns of the matrix:

$$\mathbf{N} = \begin{bmatrix} \mathbf{0}_{3\times 3} & \mathbf{C}(\mathbf{q}_{\hat{R}_0}(k'))\mathbf{g} \\ \mathbf{I}_3 & -\lfloor \mathbf{p}_{\hat{R}_0}(k') \times \rfloor \mathbf{g} \\ \mathbf{0}_{3\times 3} & -\lfloor \mathbf{v}_{\hat{R}_0}(k') \times \rfloor \mathbf{g} \\ \mathbf{0}_{3\times 3} & \mathbf{C}(\mathbf{q}_{\hat{R}_1}(k'))\mathbf{g} \\ \mathbf{I}_3 & -\lfloor \mathbf{p}_{\hat{R}_1}(k') \times \rfloor \mathbf{g} \\ \mathbf{0}_{3\times 3} & -\lfloor \mathbf{v}_{\hat{R}_1}(k') \times \rfloor \mathbf{g} \\ \vdots & \vdots \\ \mathbf{0}_{3\times 3} & \mathbf{C}(\mathbf{q}_{\hat{R}_{k'}}(k'))\mathbf{g} \\ \mathbf{I}_3 & -\lfloor \mathbf{p}_{\hat{R}_{k'}}(k') \times \rfloor \mathbf{g} \\ \mathbf{0}_{3\times 3} & -\lfloor \mathbf{v}_{\hat{R}_{k'}}(k') \times \rfloor \mathbf{g} \\ \mathbf{I}_3 & -\lfloor \mathbf{p}_{\hat{L}_1}(k') \times \rfloor \mathbf{g} \\ \mathbf{I}_3 & -\lfloor \mathbf{p}_{\hat{L}_2}(k') \times \rfloor \mathbf{g} \\ \vdots & \vdots \\ \mathbf{I}_3 & -\lfloor \mathbf{p}_{\hat{L}_n}(k') \times \rfloor \mathbf{g} \end{bmatrix} \tag{92}$$

Similarly to the case of 2D motion, the first three columns of the above matrix correspond to shifting the position of all the landmarks and all the robot positions by a constant amount (unobservable global position). The last column represents rotations of all the states about an axis parallel to the gravity vector. The physical interpretation of this result is that using only the robot-to-landmark measurements and IMU measurements, the relative positions between the robot poses and the landmarks, the robot velocity, and the robot's orientation relative to the horizontal plane can be estimated with bounded uncertainty. On the other hand, the global position of all the robot states and landmarks, as well as the global orientation about an axis parallel to the gravity vector cannot be determined.

3.3.3 Rank of $\mathbf{A}_{\text{full}}^{\text{mar}}(k')$

To compute the rank of the matrix $\mathbf{A}_{\text{full}}^{\text{mar}}(k')$ in the case of 3D motion, the analysis proceeds analogously to the analysis for the 2D case, presented in Section 3.2.3. Specifically, we can employ a decomposition of the matrix $\mathbf{M}(k,k')$ as shown in (70)-(75), and employ the following two results:

Lemma 8. *The rank of the matrix $\mathbf{K}(k,k')$ for 3D motion tracking with an IMU is $3n+6$, where n is the number of landmarks.*

Proof. See Appendix C.4. □

Lemma 9. *When the robot-to-landmark measurements are relative position meaurements, or camera measurements, and for general motion,*

$$\dim(\mathcal{N}(\mathbf{D}(k,k'))\bigcap\mathcal{R}(\mathbf{K}(k,k'))) = 0 \tag{93}$$

Proof. See Appendix C.5. □

Using the above two results, in conjunction with (76), we conclude that the robot-to-landmark measurements are camera measurements, we have:

$$\text{rank}(\mathbf{M}(k,k')) = \underbrace{\text{rank}(\mathbf{K}(k,k'))}_{3n+6} - \underbrace{\dim(\mathcal{N}(\mathbf{D}(k,k'))\bigcap\mathcal{R}(\mathbf{K}(k,k')))}_{0} = 3n+6 \tag{94}$$

Finally, employing (45), with $d_r = 9$, we get:

$$\text{rank}(\mathbf{A}_{\text{full}}^{\text{mar}}(k')) = 9k' + 3n + 6 = \text{rank}(\mathbf{A}_{\text{full}}^{\text{nm}}(k')) + 1 \tag{95}$$

This is the main result for the case of 3D motion: it shows that when the Jacobians are estimated using two different state estimates for the same variables, the rank of resulting information matrix for the entire history of states is increased. Moreover, since the size of the matrix $\mathbf{A}_{\text{full}}^{\text{mar}}(k')$ is $(9k'+3n+9)\times(9k'+3n+9)$, we see that the nullspace of the matrix is of dimension three. By inspection, we can see that the first three vectors of the matrix $\mathbf{N}$ in (92) provide a basis for this nullspace. This means that the estimator acquires erroneous information about the global orientation, the same conclusion as the one we arrive at in Section 3.2.3. Again, we emphasize that such increase in rank is incorrect since it is not justified by any new measurement information.

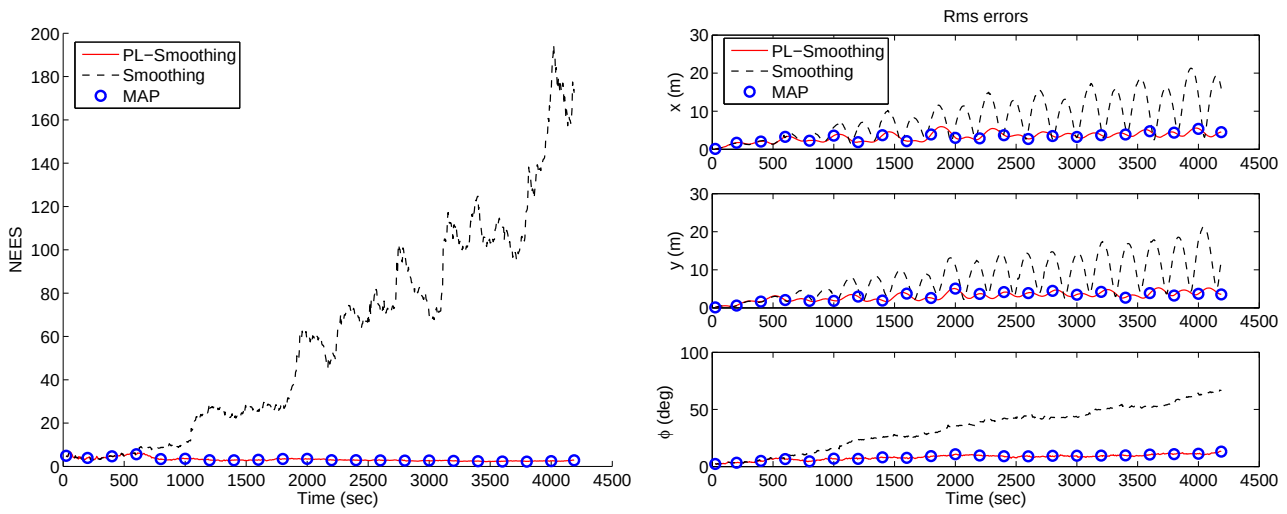
3.4 Impact on the estimator's consistency

The results of (79) and (95) show that the rank of the information matrix for the *entire* state history, $\mathbf{A}_{\text{full}}^{\text{mar}}(k')$, is incorrectly increased as a result of the use of two different estimates of some variables in computing Jacobians. Using (31) 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, $\mathbf{A}(k')$:

$$\text{rank}(\mathbf{A}(k')) = \text{rank}(\mathbf{A}_{\text{full}}^{\text{mar}}(k')) - \text{rank}(\mathbf{A}_{mm}(k)) \tag{96}$$

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 compared to its correct value, as a result of the marginalization process.

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 4.



(a) (b)

Figure 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.

3.5 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 (28), we employ the *prior* estimates, rather than the *current* ones, for states for which a prior exists. Thus, (28) is changed to:

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

In the above the only estimate of $\mathbf{x}_r$ used is $\hat{\mathbf{x}}_r(k)$. As shown above, the rank of the matrix $\mathbf{A}^{\text{mar}}_{\text{full}}(k')$ 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.

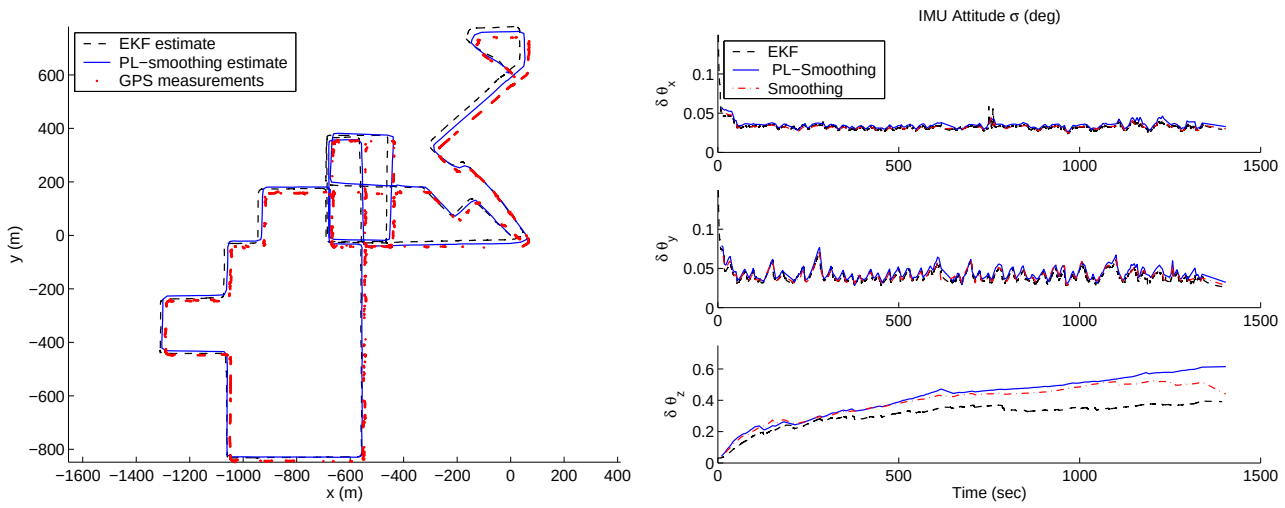
We term algorithm resulting from the above modification Prior-Linearization (PL) fixed-lag smoothing. It is important to note that, as illustrated in Fig. 1, the number of variables for which the marginalization process creates prior information is typically small. As a result, typically only a small 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 (28)) 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.

4 Results

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

4.1 Simulation results: 2D localization

In our simulation setup, we consider the case of a robot that moves on a plane along a circular trajectory of total length of approximately 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



(a) (b)

Figure 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.

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 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, while its computational cost is constant over time and orders of magnitude smaller than that of the full-state MAP.

4.2 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 tracked in each image. Features were extracted using the Harris corner detector [21], 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 [7]. 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.

Figure 3(b) shows the time evolution of the reported standard deviation for the orientation estimate. The three subplots correspond to the rotation errors about the x, y, and z axes, respectively, and the solid, dashed, and dash-dotted lines in each plot correspond to the PL-smoother, the EKF-smoother, and the SL-smoother. 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 estimate of 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 the fact 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.

5 Conclusions

In this report, we have 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), the resulting estimator becomes inconsistent. Based on our analysis, we have proposed a modified linearization scheme, termed PL-fixed lag smoothing, which ensures that no artificial information is introduced, and thus helps prevent inconsistency. The proposed algorithm was tested in both simulation and real-world experiments, and its performance was shown to be superior to that of alternative methods.

A Proof of Lemma 1

First, we note that since the matrix $\mathbf{S}$ in (35) and (38) is full-rank, we have:

$$\text{rank}\left(\mathbf{A}_{\text{full}}^{\text{nm}}(k')\right) = \text{rank}\left(\mathbf{W}(k')\right) \tag{97}$$

$$\text{rank}\left(\mathbf{A}_{\text{full}}^{\text{mar}}(k')\right) = \text{rank}\left(\mathbf{W}(k,k')\right) \tag{98}$$

We will therefore focus on the rank of the matrices $\mathbf{W}(k')$ and $\mathbf{W}(k,k')$. Since the structure of these two matrices is the same (shown in (40)), we will here drop the time indices, and provide a proof that applies to both $\mathbf{W}(k')$ and $\mathbf{W}(k,k')$.

To compute the rank of matrix $\mathbf{W}$, we first apply a number of row and column operations to transform $\mathbf{W}$ into an equivalent matrix with the same rank, but with structure that is more amenable to analysis. We will use the sign "$\sim$" to denote a transformation using row or column operations on the matrix. We have:

$$\mathbf{W} = \left[\begin{array}{cccc|c} -\mathbf{\Phi}_{R_0} & \mathbf{I}_{d_r} & \dots & \mathbf{0} & \mathbf{0} \\ \vdots & \ddots & \ddots & \vdots & \vdots \\ \mathbf{0} & \dots & -\mathbf{\Phi}_{R_{k'-1}} & \mathbf{I}_{d_r} & \mathbf{0} \\ \hline \mathbf{H}_{R_0} & \dots & \dots & \mathbf{0} & \mathbf{H}_{L_0} \\ \mathbf{0} & \ddots & \mathbf{0} & \vdots & \vdots \\ \vdots & \mathbf{0} & \ddots & \mathbf{0} & \vdots \\ \mathbf{0} & \dots & \dots & \mathbf{H}_{R_{k'}} & \mathbf{H}_{L_{k'}} \end{array}\right]$$

multiply the ith block row with $-\mathbf{\Phi}_{R_i}^{-1}$

$$\sim \left[\begin{array}{cccc|c} \mathbf{I}_{d_r} & -\mathbf{\Phi}_{R_0}^{-1} & \dots & \mathbf{0} & \mathbf{0} \\ \vdots & \ddots & \ddots & \vdots & \vdots \\ \mathbf{0} & \dots & \mathbf{I}_{d_r} & -\mathbf{\Phi}_{R_{k'-1}}^{-1} & \mathbf{0} \\ \hline \mathbf{H}_{R_0} & \dots & \dots & \mathbf{0} & \mathbf{H}_{L_0} \\ \mathbf{0} & \ddots & \mathbf{0} & \vdots & \vdots \\ \vdots & \mathbf{0} & \ddots & \mathbf{0} & \vdots \\ \mathbf{0} & \dots & \dots & \mathbf{H}_{R_{k'}} & \mathbf{H}_{L_{k'}} \end{array}\right]$$

multiply the block column corresponding to the robot pose $\mathbf{r}_{k'}$ with $(-\mathbf{\Phi}_{R_{k'-1}})$

$$\sim \left[\begin{array}{cccc|c} \mathbf{I}_{d_r} & -\mathbf{\Phi}_{R_0}^{-1} & \dots & \mathbf{0} & \mathbf{0} \\ \vdots & \ddots & \ddots & \vdots & \vdots \\ \mathbf{0} & \dots & \mathbf{I}_{d_r} & -\mathbf{I}_{d_r} & \mathbf{0} \\ \hline \mathbf{H}_{R_0} & \dots & \dots & \mathbf{0} & \mathbf{H}_{L_0} \\ \mathbf{0} & \ddots & \mathbf{0} & \vdots & \vdots \\ \vdots & \mathbf{0} & \ddots & \mathbf{0} & \vdots \\ \mathbf{0} & \dots & \mathbf{0} & \mathbf{H}_{R_{k'}}\mathbf{\Phi}_{R_{k'-1}} & \mathbf{H}_{L_{k'}} \end{array}\right]$$

add the block column of $\mathbf{r}_{k'}$ to the block column of $\mathbf{r}_{k'-1}$

$$\sim \left[\begin{array}{cccc|c} \mathbf{I}_{d_r} & -\mathbf{\Phi}_{R_0}^{-1} & \dots & \mathbf{0} & \mathbf{0} \\ \vdots & \ddots & \ddots & \vdots & \vdots \\ \mathbf{0} & \dots & \mathbf{0} & -\mathbf{I}_{d_r} & \mathbf{0} \\ \hline \mathbf{H}_{R_0} & \dots & \dots & \mathbf{0} & \mathbf{H}_{L_0} \\ \mathbf{0} & \ddots & \mathbf{0} & \vdots & \vdots \\ \vdots & \mathbf{0} & \mathbf{H}_{R_{k'-1}} & \mathbf{0} & \vdots \\ \mathbf{0} & \dots & \mathbf{H}_{R_{k'}}\mathbf{\Phi}_{R_{k'-1}} & \mathbf{H}_{R_{k'}}\mathbf{\Phi}_{R_{k'-1}} & \mathbf{H}_{L_{k'}} \end{array}\right]$$

add the k'th block row multiplied by $\mathbf{H}_{R_{k'}}\mathbf{\Phi}_{R_{k'-1}}$ to the last block row

$$\sim \left[\begin{array}{ccccc|c} \mathbf{I}_{d_r} & -\mathbf{\Phi}_{R_0}^{-1} & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \ddots & \ddots & \vdots & \mathbf{0} \\ \vdots & \vdots & \ddots & -\mathbf{I}_{d_r} & \mathbf{0} & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} & -\mathbf{I}_{d_r} & \mathbf{0} \\ \hline \mathbf{H}_{R_0} & \mathbf{0} & \dots & \dots & \mathbf{0} & \mathbf{H}_{L_0} \\ \mathbf{0} & \mathbf{H}_{R_1} & \mathbf{0} & \dots & \mathbf{0} & \vdots \\ \vdots & \vdots & \ddots & \ddots & \vdots & \mathbf{H}_{L_i} \\ \mathbf{0} & \vdots & \ddots & \mathbf{H}_{R_{k'-1}} & \mathbf{0} & \vdots \\ \mathbf{0} & \vdots & \dots & \mathbf{H}_{R_{k'}}\mathbf{\Phi}_{R_{k'-1}} & \mathbf{0} & \mathbf{H}_{L_{k'}} \end{array}\right]$$

repeating for the next columns to the left

$$\sim \left[\begin{array}{ccccc|c} & \mathbf{0} & -\mathbf{I}_{d_r} & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} \\ & \mathbf{0} & \mathbf{0} & \ddots & \ddots & \vdots & \mathbf{0} \\ & \vdots & \vdots & \ddots & -\mathbf{I}_{d_r} & \mathbf{0} & \vdots \\ & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} & -\mathbf{I}_{d_r} & \mathbf{0} \\ \hline & \mathbf{H}_{R_0} & \mathbf{0} & \dots & \dots & \mathbf{0} & \mathbf{H}_{L_0} \\ & \mathbf{H}_{R_1}\mathbf{\Phi}_{R_0} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} & \vdots \\ & \vdots & \vdots & \ddots & \ddots & \vdots & \mathbf{H}_{L_i} \\ \mathbf{H}_{R_{k'-1}}\mathbf{\Phi}_{R_{k'-2}}\dots\mathbf{\Phi}_{R_0} & & \mathbf{0} & \ddots & \mathbf{0} & \mathbf{0} & \vdots \\ \mathbf{H}_{R_{k'}}\mathbf{\Phi}_{R_{k'-1}}\dots\mathbf{\Phi}_{R_0} & & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} & \mathbf{H}_{L_{k'}} \end{array}\right]$$

$$= \left[\begin{array}{cc|c} \mathbf{0}_{d_r k' \times d_r} & -\mathbf{I}_{d_r k'} & \mathbf{0} \\ \hline \mathbf{M}_R & \mathbf{0}_{d_\ell l \times d_r k'} & \mathbf{M}_L \end{array}\right]$$

$$\sim \left[\begin{array}{cc|c} -\mathbf{I}_{d_r k'} & \mathbf{0}_{d_r k' \times d_r} & \mathbf{0} \\ \hline \mathbf{0}_{d_\ell l \times d_r k'} & \mathbf{M}_R & \mathbf{M}_L \end{array}\right]$$

$$= \left[\begin{array}{c|c} -\mathbf{I}_{d_r k'} & \mathbf{0} \\ \hline \mathbf{0}_{d_\ell l \times d_r k'} & \mathbf{M} \end{array}\right] \tag{99}$$

where $\mathbf{M} = [\mathbf{M}_R \;\; \mathbf{M}_L]$ and $\mathbf{M}_R, \mathbf{M}_L$ are the following matrices:

$$\mathbf{M}_R = \left[\begin{array}{c} \mathbf{H}_{R_0} \\ \mathbf{H}_{R_1}\mathbf{\Phi}_{R_0} \\ \vdots \\ \mathbf{H}_{R_{k'-1}}\mathbf{\Phi}_{R_{k'-2}}\dots\mathbf{\Phi}_{R_0} \\ \mathbf{H}_{R_{k'}}\mathbf{\Phi}_{R_{k'-1}}\dots\mathbf{\Phi}_{R_0} \end{array}\right], \mathbf{M}_L = \left[\begin{array}{c} \mathbf{H}_{L_0} \\ \vdots \\ \vdots \\ \mathbf{H}_{L_{k'}} \end{array}\right] \tag{100}$$

Based on the properties of partitioned matrices, we obtain from Eq. (99):

$$\text{rank}(\mathbf{W}) = \text{rank}(\mathbf{I}_{d_r k'}) + \text{rank}(\mathbf{M}) = k' d_r + \text{rank}(\mathbf{M}) \tag{101}$$

Using the above result, as well as that of (97)-(98), we obtain the results of the Lemma.

B Proofs for the case of 2D motion

B.1 Proof of Lemma 2

All the Jacobian matrices that appear in the matrix $\mathbf{K}(k')$ are estimated using the state estimates at time-step k'. Therefore, to simplify the notation, we drop the index (k') from all equations. Using Eq. (52), we obtain:

$$\mathbf{\Phi}_{R_{i+1}}\mathbf{\Phi}_{R_i} = \begin{bmatrix} \mathbf{I}_2 & \mathbf{J}(\hat{\mathbf{p}}_{R_{i+2}} - \hat{\mathbf{p}}_{R_i}) \\ \mathbf{0}_{1\times 2} & 1 \end{bmatrix} \tag{102}$$

and by induction, it is straightforward to show that for $l > 0$:

$$\mathbf{\Phi}_{R_{l-1}}\mathbf{\Phi}_{R_{l-2}}\dots\mathbf{\Phi}_{R_0} = \begin{bmatrix} \mathbf{I}_2 & \mathbf{J}(\hat{\mathbf{p}}_{R_l} - \hat{\mathbf{p}}_{R_0}) \\ \mathbf{0}_{1\times 2} & 1 \end{bmatrix} \tag{103}$$

For the i-th block of the matrix $\mathbf{K}_R$, we have:

$$\mathbf{H}'_{R_i}\mathbf{\Phi}_{R_{i-1}}\dots\mathbf{\Phi}_{R_0} = \mathbf{H}'_{R_i}\begin{bmatrix} \mathbf{I}_2 & \mathbf{J}(\hat{\mathbf{p}}_{R_i} - \hat{\mathbf{p}}_{R_0}) \\ \mathbf{0}_{1\times 2} & 1 \end{bmatrix} \tag{104}$$

$$= \begin{bmatrix} -\mathbf{I}_2 & -\mathbf{J}(\hat{\mathbf{p}}_{L_{j_1}} - \hat{\mathbf{p}}_{R_i}) \\ -\mathbf{I}_2 & -\mathbf{J}(\hat{\mathbf{p}}_{L_{j_2}} - \hat{\mathbf{p}}_{R_i}) \\ \vdots & \\ -\mathbf{I}_2 & -\mathbf{J}(\hat{\mathbf{p}}_{L_{j_{l_i}}} - \hat{\mathbf{p}}_{R_i}) \end{bmatrix} \begin{bmatrix} \mathbf{I}_2 & \mathbf{J}(\hat{\mathbf{p}}_{R_i} - \hat{\mathbf{p}}_{R_0}) \\ \mathbf{0}_{1\times 2} & 1 \end{bmatrix} \tag{105}$$

$$= \begin{bmatrix} -\mathbf{I}_2 & -\mathbf{J}(\hat{\mathbf{p}}_{L_{j_1}} - \hat{\mathbf{p}}_{R_0}) \\ -\mathbf{I}_2 & -\mathbf{J}(\hat{\mathbf{p}}_{L_{j_2}} - \hat{\mathbf{p}}_{R_0}) \\ \vdots & \\ -\mathbf{I}_2 & -\mathbf{J}(\hat{\mathbf{p}}_{L_{j_{l_i}}} - \hat{\mathbf{p}}_{R_0}) \end{bmatrix} \tag{106}$$

Substituting into the matrix $\mathbf{K}_R$, we obtain:

$$\mathbf{K}_R = \begin{bmatrix} \mathbf{H}'_{R_0} \\ \mathbf{H}'_{R_1}\mathbf{\Phi}_{R_0} \\ \vdots \\ \mathbf{H}'_{R_{k'-1}}\mathbf{\Phi}_{R_{k'-2}}\dots\mathbf{\Phi}_{R_0} \\ \mathbf{H}'_{R_{k'}}\mathbf{\Phi}_{R_{k'-1}}\dots\mathbf{\Phi}_{R_0} \end{bmatrix} = \begin{bmatrix} -\mathbf{I}_2 & -\mathbf{J}(\hat{\mathbf{p}}_{L_1} - \hat{\mathbf{p}}_{R_0}) \\ \vdots & \vdots \\ -\mathbf{I}_2 & -\mathbf{J}(\hat{\mathbf{p}}_{L_j} - \hat{\mathbf{p}}_{R_0}) \\ \vdots & \vdots \\ -\mathbf{I}_2 & -\mathbf{J}(\hat{\mathbf{p}}_{L_n} - \hat{\mathbf{p}}_{R_0}) \end{bmatrix} \tag{107}$$

Then, the matrix $\mathbf{K}$ becomes:

$$\mathbf{K} = \begin{bmatrix} \mathbf{K}_R & \mathbf{K}_L \end{bmatrix}$$

$$= \left[\begin{array}{cc|ccccc} -\mathbf{I}_2 & -\mathbf{J}(\hat{\mathbf{p}}_{L_1} - \hat{\mathbf{p}}_{R_0}) & \mathbf{I}_2 & \mathbf{0} & \dots & \dots & \mathbf{0} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ -\mathbf{I}_2 & -\mathbf{J}(\hat{\mathbf{p}}_{L_j} - \hat{\mathbf{p}}_{R_0}) & \mathbf{0} & \dots & \mathbf{I}_2 & \dots & \mathbf{0} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ -\mathbf{I}_2 & -\mathbf{J}(\hat{\mathbf{p}}_{L_n} - \hat{\mathbf{p}}_{R_0}) & \mathbf{0} & \dots & \dots & \dots & \mathbf{I}_2 \end{array}\right]$$

Add the first block column multiplied by $\mathbf{J}\hat{\mathbf{p}}_{R_0}$ to the second block column

$$\sim \left[\begin{array}{cc|ccccc} -\mathbf{I}_2 & -\mathbf{J}\hat{\mathbf{p}}_{L_1} & \mathbf{I}_2 & \mathbf{0} & \dots & \dots & \mathbf{0} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ -\mathbf{I}_2 & -\mathbf{J}\hat{\mathbf{p}}_{L_j} & \mathbf{0} & \dots & \mathbf{I}_2 & \dots & \mathbf{0} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ -\mathbf{I}_2 & -\mathbf{J}\hat{\mathbf{p}}_{L_n} & \mathbf{0} & \dots & \dots & \dots & \mathbf{I}_2 \end{array}\right]$$

Add the block column for $\mathbf{l}_j$ multiplied by $\mathbf{J}\hat{\mathbf{p}}_{L_j}$ to the second block column, for all $j = 1, \dots n$

$$\sim \left[\begin{array}{cc|ccccc} -\mathbf{I}_2 & \mathbf{0} & \mathbf{I}_2 & \mathbf{0} & \dots & \dots & \mathbf{0} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ -\mathbf{I}_2 & \mathbf{0} & \mathbf{0} & \dots & \mathbf{I}_2 & \dots & \mathbf{0} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ -\mathbf{I}_2 & \mathbf{0} & \mathbf{0} & \dots & \dots & \dots & \mathbf{I}_2 \end{array}\right]$$

Add the block column for $\mathbf{l}_j$ to the first block column, for all $j = 1, \dots n$

$$\begin{aligned} &\sim \left[\begin{array}{cc|ccccc} \mathbf{0} & \mathbf{0} & \mathbf{I}_2 & \mathbf{0} & \dots & \dots & \mathbf{0} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{I}_2 & \dots & \mathbf{0} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \dots & \dots & \dots & \mathbf{I}_2 \end{array}\right] \\ &= \begin{bmatrix} \mathbf{0}_{2l \times 3} & \mathbf{K}_L \end{bmatrix} \end{aligned} \tag{108}$$

From the above result, we conclude that the rank of the matrix $\mathbf{K}$ is equal to the rank of $\mathbf{K}_L$. Note that the matrix $\mathbf{K}_L$ has one block row for each of the available measurements $\mathbf{z}_{ij}$, and each row all zeros except for an identity matrix at the position corresponding to landmark j. Therefore, we can show that the only solution to $\mathbf{K}_L\mathbf{a} = \mathbf{0}$ is $\mathbf{a} = \mathbf{0}$, which shows that the matrix $\mathbf{K}_L$ is full column rank. Therefore,

$$\text{rank}(\mathbf{K}) = \text{rank}(\mathbf{K}_L) = 2n \tag{109}$$

and the range of the matrix $\mathbf{K}$ is spanned by the columns of $\mathbf{K}_L$:

$$\mathcal{R}(\mathbf{K}) = \mathcal{R}(\mathbf{K}_L) \tag{110}$$

B.2 Proof of Lemma 3

Using the result of (110) we can write:

$$\mathcal{N}(\mathbf{D}) \bigcap \mathcal{R}(\mathbf{K}) = \emptyset \;\Leftrightarrow\; \mathcal{N}(\mathbf{D}) \bigcap \mathcal{R}(\mathbf{K}_L) = \emptyset \;\Leftrightarrow\; \mathbf{D}\mathbf{K}_L\mathbf{a} = \mathbf{0} \text{ only when } \mathbf{a} = \mathbf{0} \tag{111}$$

Let us define the following partitioning of $\mathbf{a}$:

$$\mathbf{a} = \begin{bmatrix} \mathbf{a}_1 \\ \vdots \\ \vdots \\ \mathbf{a}_n \end{bmatrix} \tag{112}$$

where each of the $\mathbf{a}_j$ is a 2×1 vector. With this notation, and using (63) and (64), we obtain:

$$\begin{aligned} &\mathbf{D}\mathbf{K}_L\mathbf{a} = \mathbf{0} \\ \Leftrightarrow\; &\mathbf{M}_L\mathbf{a} = \mathbf{0} \\ \Leftrightarrow\; &\mathbf{H}_{L_{ij}}\mathbf{a}_j = \mathbf{0},\ \forall (i,j) \in \mathcal{S} \end{aligned}$$

which shows that to prove the lemma, we should prove

$$\mathbf{H}_{L_{ij}}\mathbf{a}_j = \mathbf{0},\ \forall (i,j) \in \mathcal{S} \Leftrightarrow \mathbf{a}_j = \mathbf{0},\ \forall j = 1, \dots, n \tag{113}$$

We now distinguish 3 cases of interest:

1) Relative position measurements: The relative position measurements are described by the following measurement function:

$$\mathbf{h}(\Delta\mathbf{p}^{ij}) = \Delta\mathbf{p}^{ij} \tag{114}$$

where $\Delta\mathbf{p}^{ij}$ is the relative robot-landmark position. For this case we obtain $\nabla\mathbf{h}_{ij} = \mathbf{I}_2$, and thus

$$\mathbf{H}_{L_{ij}} = \mathbf{C}^T(\hat{\phi}_{R_i}) \tag{115}$$

Therefore, the condition $\mathbf{H}_{L_{ij}}\mathbf{a}_j = \mathbf{0},\ \forall(i,j) \in \mathcal{S}$ implies that

$$\begin{aligned} \mathbf{C}^T(\hat{\phi}_{R_i})\mathbf{a}_j = \mathbf{0},\ \text{ for } j = 1 \dots n, \\ \Rightarrow \mathbf{a}_j = \mathbf{0}\ \text{ for } j = 1 \dots n \end{aligned}$$

Which is the desired result.

2) Relative range measurements: The relative range measurements are described by the following measurement function:

$$\mathbf{h}(\Delta\mathbf{p}^{ij}) = \mathbf{h}\left(\begin{bmatrix} \Delta p_1^{ij} \\ \Delta p_2^{ij} \end{bmatrix}\right) = \sqrt{(\Delta p_1^{ij})^2 + (\Delta p_2^{ij})^2} \tag{116}$$

where $\Delta\mathbf{p}^{ij}$ is the relative robot-landmark position. The Jacobian of $\mathbf{h}$ with respect to $\Delta\mathbf{p}^{ij}$ is:

$$\begin{aligned} \nabla\mathbf{h}_{ij} &= \begin{bmatrix} \frac{\Delta p_1^{ij}}{\sqrt{(\Delta p_1^{ij})^2+(\Delta p_2^{ij})^2}} & \frac{\Delta p_2^{ij}}{\sqrt{(\Delta p_1^{ij})^2+(\Delta p_2^{ij})^2}} \end{bmatrix} \\ &= \begin{bmatrix} \frac{\Delta p_1^{ij}}{d_{ij}} & \frac{\Delta p_2^{ij}}{d_{ij}} \end{bmatrix} \\ &= \frac{1}{d_{ij}}(\Delta\mathbf{p}^{ij})^T \end{aligned} \tag{117}$$

where $d_{ij} = \sqrt{(\Delta p_1^{ij})^2 + (\Delta p_2^{ij})^2}$. Using the definition of $\Delta\mathbf{p}^{ij}$, $\Delta\mathbf{p}^{ij} = \mathbf{C}^T(\hat{\phi}_{R_i})(\hat{\mathbf{p}}_{L_j} - \hat{\mathbf{p}}_{R_i})$, we obtain:

$$\begin{aligned} \nabla\mathbf{h}_{ij} &= \tfrac{1}{d_{ij}}(\hat{\mathbf{p}}_{L_j} - \hat{\mathbf{p}}_{R_i})^T\mathbf{C}(\hat{\phi}_{R_i}) \\ \Rightarrow \nabla\mathbf{h}_{ij}\mathbf{C}^T(\hat{\phi}_{R_i}) &= \tfrac{1}{d_{ij}}(\hat{\mathbf{p}}_{L_j} - \hat{\mathbf{p}}_{R_i})^T\mathbf{C}(\hat{\phi}_{R_i})\mathbf{C}^T(\hat{\phi}_{R_i}) \\ \Rightarrow \mathbf{H}_{L_{ij}} &= \tfrac{1}{d_{ij}}(\hat{\mathbf{p}}_{L_j} - \hat{\mathbf{p}}_{R_i})^T \end{aligned} \tag{118}$$

Therefore, the condition $\mathbf{H}_{L_{ij}}\mathbf{a}_j = \mathbf{0},\ \forall(i,j) \in \mathcal{S}$ is equivalent to:

$$\begin{aligned} (\hat{\mathbf{p}}_{L_j} - \hat{\mathbf{p}}_{R_i})^T\mathbf{a}_j = 0 \quad \forall(i,j) \in \mathcal{S} \\ \Leftrightarrow \mathbf{a}_j = \mathbf{0} \ \text{ or } \ \mathbf{a}_j \perp (\hat{\mathbf{p}}_{L_j} - \hat{\mathbf{p}}_{R_i}) \quad \forall(i,j) \in \mathcal{S} \end{aligned} \tag{119}$$

Each landmark $\mathbf{l}_j$ is generally observed from more than one robot position. Therefore, for each $\mathbf{a}_j$ there exist more than one vectors $(\hat{\mathbf{p}}_{L_j} - \hat{\mathbf{p}}_{R_i})$ that should satisfy the condition $\mathbf{a}_j = \mathbf{0}$ or $\mathbf{a}_j \perp (\hat{\mathbf{p}}_{L_j} - \hat{\mathbf{p}}_{R_i})$. Since the robot positions $\mathbf{p}_{R_i}$ are different in general, and since $(\hat{\mathbf{p}}_{L_j} - \hat{\mathbf{p}}_{R_i})$ are 2-D vectors, the only way for the conditions $\mathbf{a}_j \perp (\hat{\mathbf{p}}_{L_j} - \hat{\mathbf{p}}_{R_i})$ to be satisfied simultaneously is if all $(\hat{\mathbf{p}}_{L_j} - \hat{\mathbf{p}}_{R_i})$ are parallel, ie., if the robot moves on a straight line towards the landmark. Therefore, barring this degenerate case, the only solution to the condition $\mathbf{H}_{L_{ij}}\mathbf{a}_j = 0$ is $\mathbf{a}_j = \mathbf{0},\ \forall j$, which is the desired result.

3) Relative bearing measurements: The relative bearing measurements are described by the following measurement function:

$$\mathbf{h}(\Delta\mathbf{p}^{ij}) = \mathbf{h}\left(\begin{bmatrix} \Delta p_1^{ij} \\ \Delta p_2^{ij} \end{bmatrix}\right) = \arctan\left(\frac{\Delta p_2^{ij}}{\Delta p_1^{ij}}\right) \tag{120}$$

and the Jacobian of $\mathbf{h}$ with respect to $\Delta\mathbf{p}^{ij}$ is:

$$\begin{aligned}\nabla\mathbf{h}_{ij} &= \begin{bmatrix} \frac{-\Delta p_2^{ij}}{(\Delta p_1^{ij})^2+(\Delta p_2^{ij})^2} & \frac{\Delta p_1^{ij}}{(\Delta p_1^{ij})^2+(\Delta p_2^{ij})^2} \end{bmatrix} \\ &= \begin{bmatrix} \frac{-\Delta p_2^{ij}}{d_{ij}^2} & \frac{\Delta p_1^{ij}}{d_{ij}^2} \end{bmatrix} \\ &= \frac{1}{d_{ij}^2}(\mathbf{J}\Delta\mathbf{p}^{ij})^T \end{aligned} \tag{121}$$

where $\mathbf{J} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$. Using $\Delta\mathbf{p}^{ij} = \mathbf{C}^T(\hat{\phi}_{R_i})(\hat{\mathbf{p}}_{L_j} - \hat{\mathbf{p}}_{R_i})$ and the fact that $\mathbf{C}(\hat{\phi}_{R_i})\mathbf{J}^T = \mathbf{J}^T\mathbf{C}(\hat{\phi}_{R_i})$, we have:

$$\begin{aligned} \nabla\mathbf{h}_{ij} &= \tfrac{1}{d_{ij}^2}(\hat{\mathbf{p}}_{L_j} - \hat{\mathbf{p}}_{R_i})^T\mathbf{C}(\hat{\phi}_{R_i})\mathbf{J}^T \\ \Rightarrow \nabla\mathbf{h}_{ij}\mathbf{C}^T(\hat{\phi}_{R_i}) &= \tfrac{1}{d_{ij}^2}(\hat{\mathbf{p}}_{L_j} - \hat{\mathbf{p}}_{R_i})^T\mathbf{J}^T\mathbf{C}(\hat{\phi}_{R_i})\mathbf{C}^T(\hat{\phi}_{R_i}) \\ \Rightarrow \mathbf{H}_{L_{ij}} &= \tfrac{1}{d_{ij}^2}(\hat{\mathbf{p}}_{L_j} - \hat{\mathbf{p}}_{R_i})^T\mathbf{J}^T \end{aligned} \tag{122}$$

Therefore, the condition $\mathbf{H}_{L_{ij}}\mathbf{a}_j = \mathbf{0},\ \forall(i,j) \in \mathcal{S}$ is equivalent to:

$$\begin{aligned} &\frac{1}{d_{ij}^2}(\hat{\mathbf{p}}_{L_j} - \hat{\mathbf{p}}_{R_i})^T\mathbf{J}^T\mathbf{a}_j = \mathbf{0} \quad \forall(i,j) \in \mathcal{S} \\ &\quad\Leftrightarrow (\hat{\mathbf{p}}_{L_j} - \hat{\mathbf{p}}_{R_i})^T\mathbf{c}_j = \mathbf{0} \quad \forall(i,j) \in \mathcal{S} \\ &\quad\Leftrightarrow \mathbf{c}_j = \mathbf{0} \ \text{ or } \ \mathbf{c}_j \perp(\hat{\mathbf{p}}_{L_j} - \hat{\mathbf{p}}_{R_i}) \qquad \forall(i,j) \in \mathcal{S} \end{aligned} \tag{123}$$

where $\mathbf{c}_j = \mathbf{J}^T\mathbf{a}_j, j = 1\dots n$. Note that the above condition for $\mathbf{c}_j$ is identical to the condition derived for $\mathbf{a}_j$ in the case of relative range measurements. Therefore, by the same argument, we conclude that barring the degenerate case where the robot moves on a straight line towards the landmark, we must have $\mathbf{c}_j = \mathbf{0}$, which directly implies that $\mathbf{a}_j = \mathbf{0}$, given that $\mathbf{J}$ is an invertible matrix.

Conclusion: In summary, we conclude that when the robot-to-landmark measurements are relative positions, relative ranges, or relative bearings:

$$\mathcal{N}(\mathbf{D})\bigcap\mathcal{R}(\mathbf{K}) = \emptyset \Rightarrow \dim(\mathcal{N}(\mathbf{D})\bigcap\mathcal{R}(\mathbf{K})) = 0 \tag{124}$$

B.3 Proof of Lemma 4

For the proof of this result we will follow a course similar to that of Appendix B.1. First, we note that all state transition matrices $\mathbf{\Phi}_{R_i}$, for $i = 0\dots m-1$ are evaluated using the state estimates available at time-step k, while those for $i \geq m$ are evaluated using the state estimates at time-step k'. Therefore, while we can write expressions analogous to (102) for $i < m-1$ and $i > m-1$, for $i = m-1$ we obtain:

$$\mathbf{\Phi}_{R_m}(k')\mathbf{\Phi}_{R_{m-1}}(k) = \begin{bmatrix} \mathbf{I}_2 & \mathbf{J}(\hat{\mathbf{p}}_{R_{m+1}}(k') - \hat{\mathbf{p}}_{R_{m-1}}(k) - \Delta\mathbf{p}_m) \\ \mathbf{0} & 1 \end{bmatrix} \tag{125}$$

where $\Delta\mathbf{p}_m = \hat{\mathbf{p}}_{R_m}(k') - \hat{\mathbf{p}}_{R_m}(k)$. Using this result, we can show that

$$\begin{aligned} \mathbf{\Phi}_{R_{l-1}}\mathbf{\Phi}_{R_{l-2}}\dots\mathbf{\Phi}_{R_0} &= \begin{bmatrix} \mathbf{I}_2 & \mathbf{J}(\hat{\mathbf{p}}_{R_l}(k) - \hat{\mathbf{p}}_{R_0}(k)) \\ \mathbf{0}_{1\times 2} & 1 \end{bmatrix},\ l \leq m \\ \mathbf{\Phi}_{R_{l-1}}\mathbf{\Phi}_{R_{l-2}}\dots\mathbf{\Phi}_{R_0} &= \begin{bmatrix} \mathbf{I}_2 & \mathbf{J}(\hat{\mathbf{p}}_{R_l}(k') - \hat{\mathbf{p}}_{R_0}(k) - \Delta\mathbf{p}_m) \\ \mathbf{0}_{1\times 2} & 1 \end{bmatrix},\ l > m \end{aligned} \tag{126}$$

Substituting into the matrix $\mathbf{K}_R(k,k')$, we obtain:

$$\mathbf{K}_R(k,k') = \left[\begin{array}{c} \mathbf{H}'_{R_0}(k) \\ \mathbf{H}'_{R_1}(k)\mathbf{\Phi}_{R_0}(k) \\ \vdots \\ \mathbf{H}'_{R_{m-1}}(k)\mathbf{\Phi}_{R_{m-2}}(k)\dots\mathbf{\Phi}_{R_0}(k) \\ \hline \mathbf{H}'_{R_m}(k')\mathbf{\Phi}_{R_{m-1}}(k)\dots\mathbf{\Phi}_{R_0}(k) \\ \mathbf{H}'_{R_{m+1}}(k')\mathbf{\Phi}_{R_m}(k')\dots\mathbf{\Phi}_{R_0}(k) \\ \vdots \\ \mathbf{H}'_{R_{k'-1}}(k')\mathbf{\Phi}_{R_{k'-2}}(k')\dots\mathbf{\Phi}_{R_0}(k) \\ \mathbf{H}'_{R_{k'}}(k')\mathbf{\Phi}_{R_{k-1}}(k')\dots\mathbf{\Phi}_{R_0}(k) \end{array}\right] = \left[\begin{array}{cc} -\mathbf{I}_2 & -\mathbf{J}(\hat{\mathbf{p}}_{L_1}(k) - \hat{\mathbf{p}}_{R_0}(k)) \\ \vdots & \vdots \\ \vdots & \vdots \\ -\mathbf{I}_2 & -\mathbf{J}(\hat{\mathbf{p}}_{L_a}(k) - \hat{\mathbf{p}}_{R_0}(k)) \\ \hline -\mathbf{I}_2 & -\mathbf{J}(\hat{\mathbf{p}}_{L_a}(k') - \hat{\mathbf{p}}_{R_0}(k) - \Delta\mathbf{p}_m) \\ \vdots & \vdots \\ \vdots & \vdots \\ -\mathbf{I}_2 & -\mathbf{J}(\hat{\mathbf{p}}_{L_n}(k') - \hat{\mathbf{p}}_{R_0}(k) - \Delta\mathbf{p}_m) \end{array}\right] \tag{127}$$

where the landmark $\mathbf{l}_a$ is one seen both from a marginalized robot pose, *and* from a robot pose that remained active after the marginalization. Substituting in the matrix $\mathbf{K}(k,k')$, we obtain:

$$\begin{aligned} \mathbf{K}(k,k') &= \begin{bmatrix} \mathbf{K}_R(k,k') & \mathbf{K}_L \end{bmatrix} \\ &= \left[\begin{array}{cc|ccccc} -\mathbf{I}_2 & -\mathbf{J}(\hat{\mathbf{p}}_{L_1}(k) - \hat{\mathbf{p}}_{R_0}(k)) & \mathbf{I}_2 & \mathbf{0}_{2\times2} & \dots & \dots & \mathbf{0}_{2\times2} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ -\mathbf{I}_2 & -\mathbf{J}(\hat{\mathbf{p}}_{L_a}(k) - \hat{\mathbf{p}}_{R_0}(k)) & \mathbf{0}_{2\times2} & \dots & \mathbf{I}_2 & \dots & \mathbf{0}_{2\times2} \\ \hline -\mathbf{I}_2 & -\mathbf{J}(\hat{\mathbf{p}}_{L_a}(k') - \hat{\mathbf{p}}_{R_0}(k) - \Delta\mathbf{p}_m) & \mathbf{0}_{2\times2} & \dots & \mathbf{I}_2 & \dots & \mathbf{0}_{2\times2} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ -\mathbf{I}_2 & -\mathbf{J}(\hat{\mathbf{p}}_{L_n}(k') - \hat{\mathbf{p}}_{R_0}(k) - \Delta\mathbf{p}_m) & \mathbf{0}_{2\times2} & \dots & \dots & \dots & \mathbf{I}_2 \end{array}\right] \end{aligned}$$

Add the first block column multiplied by $\mathbf{J}\hat{\mathbf{p}}_{R_0}(k)$ to the second block column

$$\sim \left[\begin{array}{cc|ccccc} -\mathbf{I}_2 & -\mathbf{J}(\hat{\mathbf{p}}_{L_1}(k)) & \mathbf{I}_2 & \mathbf{0}_{2\times2} & \dots & \dots & \mathbf{0}_{2\times2} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ -\mathbf{I}_2 & -\mathbf{J}(\hat{\mathbf{p}}_{L_a}(k)) & \mathbf{0}_{2\times2} & \dots & \mathbf{I}_2 & \dots & \mathbf{0}_{2\times2} \\ \hline -\mathbf{I}_2 & -\mathbf{J}(\hat{\mathbf{p}}_{L_a}(k') - \Delta\mathbf{p}_m) & \mathbf{0}_{2\times2} & \dots & \mathbf{I}_2 & \dots & \mathbf{0}_{2\times2} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ -\mathbf{I}_2 & -\mathbf{J}(\hat{\mathbf{p}}_{L_n}(k') - \Delta\mathbf{p}_m) & \mathbf{0}_{2\times2} & \dots & \dots & \dots & \mathbf{I}_2 \end{array}\right]$$

Add the block column for $\mathbf{l}_j$ to the first block column, for all $j = 1, \dots n$

$$\sim \left[\begin{array}{cc|ccccc} \mathbf{0}_{2\times2} & -\mathbf{J}(\hat{\mathbf{p}}_{L_1}(k)) & \mathbf{I}_2 & \mathbf{0}_{2\times2} & \dots & \dots & \mathbf{0}_{2\times2} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{0}_{2\times2} & -\mathbf{J}(\hat{\mathbf{p}}_{L_a}(k)) & \mathbf{0} & \dots & \mathbf{I}_2 & \dots & \mathbf{0}_{2\times2} \\ \hline \mathbf{0}_{2\times2} & -\mathbf{J}(\hat{\mathbf{p}}_{L_a}(k') - \Delta\mathbf{p}_m) & \mathbf{0}_{2\times2} & \dots & \mathbf{I}_2 & \dots & \mathbf{0}_{2\times2} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{0}_{2\times2} & -\mathbf{J}(\hat{\mathbf{p}}_{L_n}(k') - \Delta\mathbf{p}_m) & \mathbf{0}_{2\times2} & \dots & \dots & \dots & \mathbf{I}_2 \end{array}\right]$$

$$\sim \left[\begin{array}{cc|ccccc} \mathbf{0}_{2\times2} & \mathbf{0}_{2\times1} & \mathbf{I}_2 & \mathbf{0}_{2\times2} & \dots & \dots & \mathbf{0}_{2\times2} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{0}_{2\times2} & \mathbf{J}\big(\hat{\mathbf{p}}_{L_a}(k') - \hat{\mathbf{p}}_{L_a}(k) - \Delta\mathbf{p}_m\big) & \mathbf{0}_{2\times2} & \dots & \mathbf{I}_2 & \dots & \mathbf{0}_{2\times2} \\ \hline \mathbf{0}_{2\times2} & \mathbf{0}_{2\times1} & \mathbf{0}_{2\times2} & \dots & \mathbf{I}_2 & \dots & \mathbf{0}_{2\times2} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{0}_{2\times2} & \mathbf{0}_{2\times1} & \mathbf{0}_{2\times2} & \dots & \dots & \dots & \mathbf{I}_2 \end{array}\right]$$

$$= \begin{bmatrix} \mathbf{0}_{2l\times2} & \mathbf{k}_\phi & | & \mathbf{K}_L \end{bmatrix} \tag{128}$$

where $\mathbf{k}_\phi$ denotes the third column of the above matrix. The last equivalence relationship is obtained by the following column operations: we multiply the block columns corresponding those landmarks $\mathbf{l}_j$ that are seen *only* by the marginalized robot poses, each by $\mathbf{J}\hat{\mathbf{p}}_{L_j}(k)$, and add to the second block column. Then, we multiply the block columns corresponding

to all other landmarks $\mathbf{l}_j$, each by $\mathbf{J}\left(\hat{\mathbf{p}}_{L_j}(k') - \Delta\mathbf{p}_m\right)$, and add to the second block column. Thus, for those landmarks that are seen by the marginalized robot poses *only* or the remaining robot poses *only*, the first block column is reduced to $\mathbf{0}_{2\times 1}$. The non-zero block rows correspond to those measurements between marginalized robot poses and landmarks that are seen by both marginalized robot poses and non-marginalized robot poses.

From the above equivalence relations, we conclude that

$$\text{rank}(\mathbf{K}(k,k')) = \text{rank}\left(\begin{bmatrix}\mathbf{k}_\phi & \mathbf{K}_L\end{bmatrix}\right)$$

To evaluate the rank of the $2l \times (2n+1)$ matrix $\begin{bmatrix}\mathbf{k}_\phi & \mathbf{K}_L\end{bmatrix}$, we will compute all vectors $\mathbf{a}$ such that $\begin{bmatrix}\mathbf{k}_\phi & \mathbf{K}_L\end{bmatrix}\mathbf{a} = \mathbf{0}$. Since $\begin{bmatrix}\mathbf{k}_\phi & \mathbf{K}_L\end{bmatrix}$ has $2n+1$ columns, we introduce the following partitioning for the vector $\mathbf{a}$:

$$\mathbf{a} = \begin{bmatrix} a_\phi \\ \mathbf{a}_1 \\ \mathbf{a}_2 \\ \vdots \\ \mathbf{a}_n \end{bmatrix} \tag{129}$$

where a_ϕ is a scalar, and $\mathbf{a}_j$, $j = 1 \dots n$ are 2×1 vectors. Based on this partitioning, the product $\begin{bmatrix}\mathbf{k}_\phi & \mathbf{K}_L\end{bmatrix}\mathbf{a}$ can be written as:

$$\left[\begin{array}{c|ccccc} \mathbf{0}_{2\times 1} & \mathbf{I}_2 & \mathbf{0}_{2\times 2} & \dots & \dots & \mathbf{0}_{2\times 2} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{J}\left(\hat{\mathbf{p}}_{L_a}(k') - \hat{\mathbf{p}}_{L_a}(k) - \Delta\mathbf{p}_m\right) & \mathbf{0}_{2\times 2} & \dots & \mathbf{I}_2 & \dots & \mathbf{0}_{2\times 2} \\ \hline \mathbf{0}_{2\times 1} & \mathbf{0}_{2\times 2} & \dots & \mathbf{I}_2 & \dots & \mathbf{0}_{2\times 2} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{0}_{2\times 1} & \mathbf{0}_{2\times 2} & \dots & \dots & \dots & \mathbf{I}_2 \end{array}\right] \begin{bmatrix} a_\phi \\ \mathbf{a}_1 \\ \mathbf{a}_2 \\ \vdots \\ \mathbf{a}_n \end{bmatrix} = \begin{bmatrix} \mathbf{a}_1 \\ \vdots \\ \mathbf{J}\left(\hat{\mathbf{p}}_{L_a}(k') - \hat{\mathbf{p}}_{L_a}(k) - \Delta\mathbf{p}_m\right)a_\phi + \mathbf{a}_a \\ \mathbf{a}_a \\ \vdots \\ \mathbf{a}_n \end{bmatrix}$$

We therefore see that for the condition $\begin{bmatrix}\mathbf{k}_\phi & \mathbf{K}_L\end{bmatrix}\mathbf{a} = \mathbf{0}$ to hold, all the vectors $\mathbf{a}_j$ corresponding to landmarks that are observed *only* from marginalized robot poses, as well as landmarks observed by any non-marginalized robot poses (and possibly *also* from marginalized ones), must equal zero. Clearly, this means that *all* the vectors $\mathbf{a}_j$, $j = 1 \dots n$ must equal zero. Thus the only remaining variable to determine is a_ϕ. Using $\mathbf{a}_j = 0$, $j = 1 \dots n$, the condition $\begin{bmatrix}\mathbf{k}_\phi & \mathbf{K}_L\end{bmatrix}\mathbf{a} = \mathbf{0}$ results in the equation:

$$\mathbf{J}\left(\hat{\mathbf{p}}_{L_a}(k') - \hat{\mathbf{p}}_{L_a}(k) - \Delta\mathbf{p}_m\right)a_\phi = \mathbf{0}_{2\times 1} \tag{130}$$

Since in general the term $\hat{\mathbf{p}}_{L_a}(k') - \hat{\mathbf{p}}_{L_a}(k) - \Delta\mathbf{p}_m$ is different than zero (i.e., in general the correction in the landmark position is different than the correction in the position of the robot at time-step m), the above condition can only be satisfied for $a_\phi = 0$.

We have thus shown that the condition $\begin{bmatrix}\mathbf{k}_\phi & \mathbf{K}_L\end{bmatrix}\mathbf{a} = \mathbf{0}$ is satisfied only if $\mathbf{a} = \mathbf{0}$. This proves that $\begin{bmatrix}\mathbf{k}_\phi & \mathbf{K}_L\end{bmatrix}$ is full column rank, i.e., $\text{rank}(\begin{bmatrix}\mathbf{k}_\phi & \mathbf{K}_L\end{bmatrix}) = 2n+1$. Since $\text{rank}(\mathbf{K}(k,k')) = \text{rank}(\begin{bmatrix}\mathbf{k}_\phi & \mathbf{K}_L\end{bmatrix})$, we conclude that $\text{rank}(\mathbf{K}(k,k')) = 2n+1$, which is the desired result.

B.4 Proof of Lemma 5

In Appendix B.3, we proved that the matrix $\mathbf{K}(k,k')$ has the same rank as the full-column-rank matrix $\begin{bmatrix}\mathbf{k}_\phi & \mathbf{K}_L\end{bmatrix}$. Therefore, the range of $\mathbf{K}(k,k')$ is spanned by the columns of $\begin{bmatrix}\mathbf{k}_\phi & \mathbf{K}_L\end{bmatrix}$. Therefore, we can write:

$$\begin{aligned} &\mathcal{N}(\mathbf{D}(k,k')) \bigcap \mathcal{R}(\mathbf{K}(k,k')) = \emptyset \\ \Leftrightarrow\ &\mathcal{N}(\mathbf{D}(k,k')) \bigcap \mathcal{R}([\mathbf{k}_\phi \ \ \mathbf{K}_L]) = \emptyset \\ \Leftrightarrow\ &\mathbf{D}(k,k')[\mathbf{k}_\phi \ \ \mathbf{K}_L]\mathbf{a} = \mathbf{0} \text{ only when } \mathbf{a} = \mathbf{0} \end{aligned} \tag{131}$$

We now examine 3 cases of interest:

1) Relative position measurements: When the robot-to-landmark measurements are relative-position measurements, then the matrix $\mathbf{D}(k,k')$ is invertible, since it is a block diagonal matrix containing rotation matrices in its diagonal (see (71) and (115)). Therefore, we have

$$\mathbf{D}(k,k')[\mathbf{k}_\phi \;\; \mathbf{K}_L]\mathbf{a} = \mathbf{0} \Leftrightarrow [\mathbf{k}_\phi \;\; \mathbf{K}_L]\mathbf{a} = \mathbf{0} \Leftrightarrow \mathbf{a} = \mathbf{0} \tag{132}$$

where the last equivalence comes from the fact that the the matrix $[\mathbf{k}_\phi \;\; \mathbf{K}_L]$ is full column rank, as shown in Appendix B.3.

2) Relative range measurements: Using the expression for the Jacobian of the relative-range measurements in (117), as well as a partitioning for the vector $\mathbf{a}$ as in (129), we obtain:

$$\begin{aligned}
\mathbf{D}(k,k')[\mathbf{k}_\phi \;\; \mathbf{K}_L]\mathbf{a} &= \mathbf{D}(k,k') \left[\begin{array}{c|cccccc} \mathbf{0}_{2\times 1} & \mathbf{I}_2 & \mathbf{0}_{2\times 2} & \dots & \dots & \mathbf{0}_{2\times 2} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{J}\big(\hat{\mathbf{p}}_{L_a}(k') - \hat{\mathbf{p}}_{L_a}(k) - \Delta\mathbf{p}_m\big) & \mathbf{0}_{2\times 2} & \dots & \mathbf{I}_2 & \dots & \mathbf{0}_{2\times 2} \\ \hline \mathbf{0}_{2\times 1} & \mathbf{0}_{2\times 2} & \dots & \mathbf{I}_2 & \dots & \mathbf{0}_{2\times 2} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{0}_{2\times 1} & \mathbf{0}_{2\times 2} & \dots & \dots & \dots & \mathbf{I}_2 \end{array}\right] \begin{bmatrix} a_\phi \\ \mathbf{a}_1 \\ \mathbf{a}_2 \\ \vdots \\ \mathbf{a}_n \end{bmatrix} \\
&= \mathbf{D}(k,k') \begin{bmatrix} \mathbf{a}_1 \\ \vdots \\ \mathbf{J}\big(\hat{\mathbf{p}}_{L_a}(k') - \hat{\mathbf{p}}_{L_a}(k) - \Delta\mathbf{p}_m\big)a_\phi + \mathbf{a}_a \\ \mathbf{a}_a \\ \vdots \\ \mathbf{a}_n \end{bmatrix} \\
&= \begin{bmatrix} \frac{1}{d_{11}}\left(\hat{\mathbf{p}}_{L_1}(k) - \hat{\mathbf{p}}_{R_1}(k)\right)^T \mathbf{a}_1 \\ \vdots \\ \frac{1}{d_{(m-1)a}}\left(\hat{\mathbf{p}}_{L_a}(k) - \hat{\mathbf{p}}_{R_{m-1}}(k)\right)^T \left(\mathbf{J}\big(\hat{\mathbf{p}}_{L_a}(k') - \hat{\mathbf{p}}_{L_a}(k) - \Delta\mathbf{p}_m\big)a_\phi + \mathbf{a}_a\right) \\ \frac{1}{d_{(m-1)a}}\left(\hat{\mathbf{p}}_{L_a}(k') - \hat{\mathbf{p}}_{R_{m-1}}(k')\right)^T \mathbf{a}_a \\ \vdots \\ \frac{1}{d_{k'n}}\left(\hat{\mathbf{p}}_{L_n}(k') - \hat{\mathbf{p}}_{R_{k'}}(k')\right)^T \mathbf{a}_n \end{bmatrix}
\end{aligned} \tag{133, 134}$$

Therefore, the condition $\mathbf{D}(k,k')[\mathbf{k}_\phi \;\; \mathbf{K}_L]\mathbf{a} = \mathbf{0}$ is equivalent to the conditions:

$$\begin{aligned}
\left(\hat{\mathbf{p}}_{L_1}(k) - \hat{\mathbf{p}}_{R_1}(k)\right)^T \mathbf{a}_1 &= 0 \\
&\vdots \\
\left(\hat{\mathbf{p}}_{L_a}(k) - \hat{\mathbf{p}}_{R_{m-1}}(k)\right)^T \left(\mathbf{J}\big(\hat{\mathbf{p}}_{L_a}(k') - \hat{\mathbf{p}}_{L_a}(k) - \Delta\mathbf{p}_m\big)a_\phi + \mathbf{a}_a\right) &= 0 \\
\left(\hat{\mathbf{p}}_{L_a}(k') - \hat{\mathbf{p}}_{R_{m-1}}(k')\right)^T \mathbf{a}_a &= 0 \\
&\vdots \\
\left(\hat{\mathbf{p}}_{L_n}(k') - \hat{\mathbf{p}}_{R_{k'}}(k')\right)^T \mathbf{a}_n &= 0
\end{aligned}$$

By following an argument similar to that in Appendix B.2, we can show that barring degenerate cases, the conditions $\left(\hat{\mathbf{p}}_{L_j}(k) - \hat{\mathbf{p}}_{R_i}(k)\right)^T \mathbf{a}_j = 0$ can be satisfied only if all the vectors $\mathbf{a}_j$, $j = 1 \dots n$ are equal to zero. Given these, we obtain the condition for a_ϕ:

$$\left(\hat{\mathbf{p}}_{L_a}(k) - \hat{\mathbf{p}}_{R_{m-1}}(k)\right)^T \mathbf{J}\big(\hat{\mathbf{p}}_{L_a}(k') - \hat{\mathbf{p}}_{L_a}(k) - \Delta\mathbf{p}_m\big)a_\phi = 0$$

Since in general the term $\left(\hat{\mathbf{p}}_{L_a}(k) - \hat{\mathbf{p}}_{R_{m-1}}(k)\right)^T \mathbf{J}\big(\hat{\mathbf{p}}_{L_a}(k') - \hat{\mathbf{p}}_{L_a}(k) - \Delta\mathbf{p}_m\big)$ is different than zero, we conclude that $a_\phi = 0$.

We have thus shown that the condition $\mathbf{D}(k,k')[\mathbf{k}_\phi \;\; \mathbf{K}_L]\mathbf{a} = \mathbf{0}$ is satisfied only if $\mathbf{a} = \mathbf{0}$, which is the desired result.

3) Relative bearing measurements: Using the expression for the Jacobian of the relative-bearing measurements in (121), the vector $\mathbf{a}$ shown in (129), and following steps similar to the case of relative-range measurements above, we can show that the condition $\mathbf{D}(k,k')[\mathbf{k}_\phi \;\; \mathbf{K}_L]\mathbf{a} = \mathbf{0}$ in this case is equivalent to the conditions:

$$\begin{aligned}
\big(\hat{\mathbf{p}}_{L_1}(k) - \hat{\mathbf{p}}_{R_1}(k)\big)^T \mathbf{J}\mathbf{a}_1 &= 0 \\
&\vdots \\
\big(\hat{\mathbf{p}}_{L_a}(k) - \hat{\mathbf{p}}_{R_{m-1}}(k)\big)^T \mathbf{J}\big(\mathbf{J}\big(\hat{\mathbf{p}}_{L_a}(k') - \hat{\mathbf{p}}_{L_a}(k) - \Delta\mathbf{p}_m\big)a_\phi + \mathbf{a}_a\big) &= 0 \\
\big(\hat{\mathbf{p}}_{L_a}(k') - \hat{\mathbf{p}}_{R_{m-1}}(k')\big)^T \mathbf{J}\mathbf{a}_a &= 0 \\
&\vdots \\
\big(\hat{\mathbf{p}}_{L_n}(k') - \hat{\mathbf{p}}_{R_{k'}}(k')\big)^T \mathbf{J}\mathbf{a}_n &= 0
\end{aligned}$$

By setting $\mathbf{c}_j = \mathbf{J}\mathbf{a}_j$, the above expressions become identical to those in the previous case of relative-range measurements. Therefore, we can show that barring degenerate cases, the only possible solution is $\mathbf{a} = \mathbf{0}$.

Conclusion: In summary, we conclude that when the robot-to-landmark measurements are relative positions, relative ranges, or relative bearings:

$$\mathcal{N}(\mathbf{D}(k,k')) \bigcap \mathcal{R}(\mathbf{K}(k,k')) = \emptyset \Rightarrow \dim(\mathcal{N}(\mathbf{D}(k,k')) \bigcap \mathcal{R}(\mathbf{K}(k,k'))) = 0 \tag{135}$$

C Proofs for the case of 3D motion

C.1 Proof of Lemma 6

The proof of this result follows a course similar to the proof of Lemma 2 in the 2D motion case. Similarly to that case, all quantities appearing in this proof are evaluated using the estimates at time step k', and therefore we omit the time index (k'), for notation simplicity.

Using the definition of the matrix $\mathbf{\Phi}_{R_i}$ in. (82), we can show that:

$$\mathbf{\Phi}_{R_{i+1}}\mathbf{\Phi}_{R_i} = \begin{bmatrix} {}^{R_{i+2}}_{R_i}\mathbf{R} & \mathbf{0} & \mathbf{0} \\ -\lfloor(\hat{\mathbf{p}}_{R_{i+2}} - \hat{\mathbf{p}}_{R_i} - \hat{\mathbf{v}}_{R_i}({}^{i+2}_{i}\Delta t) - \frac{\mathbf{g}}{2}({}^{i+2}_{i}\Delta t)^2)\times\rfloor \mathbf{C}^T(\hat{\mathbf{q}}_{R_i}) & \mathbf{I}_3 & {}^{i+2}_{i}\Delta t\mathbf{I}_3 \\ -\lfloor(\hat{\mathbf{v}}_{R_{i+2}} - \hat{\mathbf{v}}_{R_i} - \mathbf{g}({}^{i+2}_{i}\Delta t))\times\rfloor \mathbf{C}^T(\hat{\mathbf{q}}_{R_i}) & \mathbf{0} & \mathbf{I}_3 \end{bmatrix} \tag{136}$$

and by induction, we can show that for $l > 0$:

$$\mathbf{\Phi}_{R_{l-1}}\dots\mathbf{\Phi}_{R_0} = \begin{bmatrix} {}^{R_l}_{R_0}\mathbf{R} & \mathbf{0} & \mathbf{0} \\ -\lfloor(\hat{\mathbf{p}}_{R_l} - \hat{\mathbf{p}}_{R_0} - \hat{\mathbf{v}}_{R_0}({}^{l}_{0}\Delta t) - \frac{\mathbf{g}}{2}({}^{l}_{0}\Delta t)^2)\times\rfloor \mathbf{C}^T(\hat{\mathbf{q}}_{R_0}) & \mathbf{I}_3 & {}^{l}_{0}\Delta t\mathbf{I}_3 \\ -\lfloor(\hat{\mathbf{v}}_{R_l} - \hat{\mathbf{v}}_{R_0} - \mathbf{g}({}^{l}_{0}\Delta t))\times\rfloor \mathbf{C}^T(\hat{\mathbf{q}}_{R_0}) & \mathbf{0} & \mathbf{I}_3 \end{bmatrix} \tag{137}$$

Using the above result, and the definition of the matrix $\mathbf{H}'_{R_i}$ in (87), we obtain:

$$\mathbf{H}'_{R_i}\mathbf{\Phi}_{R_{i-1}}\dots\mathbf{\Phi}_{R_0} = \begin{bmatrix} \lfloor(\hat{\mathbf{p}}_{L_{j_1}} - \hat{\mathbf{p}}_{R_0} - \hat{\mathbf{v}}_{R_0}({}^{i}_{0}\Delta t) - \frac{\mathbf{g}}{2}({}^{i}_{0}\Delta t)^2)\times\rfloor \mathbf{C}^T(\hat{\mathbf{q}}_{R_0}) & -\mathbf{I}_3 & -{}^{i}_{0}\Delta t\mathbf{I}_3 \\ \lfloor(\hat{\mathbf{p}}_{L_{j_2}} - \hat{\mathbf{p}}_{R_0} - \hat{\mathbf{v}}_{R_0}({}^{i}_{0}\Delta t) - \frac{\mathbf{g}}{2}({}^{i}_{0}\Delta t)^2)\times\rfloor \mathbf{C}^T(\hat{\mathbf{q}}_{R_0}) & -\mathbf{I}_3 & -{}^{i}_{0}\Delta t\mathbf{I}_3 \\ \vdots & \vdots & \vdots \\ \lfloor(\hat{\mathbf{p}}_{L_{j_{l_i}}} - \hat{\mathbf{p}}_{R_0} - \hat{\mathbf{v}}_{R_0}({}^{i}_{0}\Delta t) - \frac{\mathbf{g}}{2}({}^{i}_{0}\Delta t)^2)\times\rfloor \mathbf{C}^T(\hat{\mathbf{q}}_{R_0}) & -\mathbf{I}_3 & -{}^{i}_{0}\Delta t\mathbf{I}_3 \end{bmatrix} \tag{138}$$

Substituting into the matrix $\mathbf{K}_R$, we obtain:

$$\mathbf{K}_R = \begin{bmatrix} \mathbf{H}'_{R_0} \\ \mathbf{H}'_{R_1}\mathbf{\Phi}_{R_0} \\ \vdots \\ \mathbf{H}'_{R_{k'-1}}\mathbf{\Phi}_{R_{k'-2}}\dots\mathbf{\Phi}_{R_0} \\ \mathbf{H}'_{R_{k'}}\mathbf{\Phi}_{R_{k'-1}}\dots\mathbf{\Phi}_{R_0} \end{bmatrix}$$

$$= \begin{bmatrix} \lfloor(\hat{\mathbf{p}}_{L_1} - \hat{\mathbf{p}}_{R_0} - \hat{\mathbf{v}}_{R_0}({}^0_0\Delta t) - \frac{\mathbf{g}}{2}({}^0_0\Delta t)^2)\times\rfloor \mathbf{C}^T(\hat{\mathbf{q}}_{R_0}) & -\mathbf{I}_3 & -{}^0_0\Delta t\mathbf{I}_3 \\ \vdots & \vdots & \vdots \\ \lfloor(\hat{\mathbf{p}}_{L_j} - \hat{\mathbf{p}}_{R_0} - \hat{\mathbf{v}}_{R_0}({}^i_0\Delta t) - \frac{\mathbf{g}}{2}({}^i_0\Delta t)^2)\times\rfloor \mathbf{C}^T(\hat{\mathbf{q}}_{R_0}) & -\mathbf{I}_3 & -{}^i_0\Delta t\mathbf{I}_3 \\ \vdots & \vdots & \vdots \\ \lfloor(\hat{\mathbf{p}}_{L_N} - \hat{\mathbf{p}}_{R_0} - \hat{\mathbf{v}}_{R_0}({}^{k'}_0\Delta t) - \frac{\mathbf{g}}{2}({}^{k'}_0\Delta t)^2)\times\rfloor \mathbf{C}^T(\hat{\mathbf{q}}_{R_0}) & -\mathbf{I}_3 & -{}^{k'}_0\Delta t\mathbf{I}_3 \end{bmatrix} \tag{139}$$

Therefore the matrix $\mathbf{K}$ is equal to:

$$\mathbf{K} = \left[\begin{array}{ccc|ccccc} \lfloor(\hat{\mathbf{p}}_{L_1} - \hat{\mathbf{p}}_{R_0} - \hat{\mathbf{v}}_{R_0}({}^0_0\Delta t) - \frac{\mathbf{g}}{2}({}^0_0\Delta t)^2)\times\rfloor \mathbf{C}^T(\hat{\mathbf{q}}_{R_0}) & -\mathbf{I}_3 & -{}^0_0\Delta t\mathbf{I}_3 & \mathbf{I}_3 & \mathbf{0}_{3\times3} & \dots & \dots & \mathbf{0}_{3\times3} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \lfloor(\hat{\mathbf{p}}_{L_j} - \hat{\mathbf{p}}_{R_0} - \hat{\mathbf{v}}_{R_0}({}^i_0\Delta t) - \frac{\mathbf{g}}{2}({}^i_0\Delta t)^2)\times\rfloor \mathbf{C}^T(\hat{\mathbf{q}}_{R_0}) & -\mathbf{I}_3 & -{}^i_0\Delta t\mathbf{I}_3 & \mathbf{0}_{3\times3} & \dots & \mathbf{I}_3 & \dots & \mathbf{0}_{3\times3} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \lfloor(\hat{\mathbf{p}}_{L_N} - \hat{\mathbf{p}}_{R_0} - \hat{\mathbf{v}}_{R_0}({}^{k'}_0\Delta t) - \frac{\mathbf{g}}{2}({}^{k'}_0\Delta t)^2)\times\rfloor \mathbf{C}^T(\hat{\mathbf{q}}_{R_0}) & -\mathbf{I}_3 & -{}^{k'}_0\Delta t\mathbf{I}_3 & \mathbf{0}_{3\times3} & \dots & \dots & \dots & \mathbf{I}_3 \end{array}\right]$$

Add the block columns for $\mathbf{l}_j$ multiplied by $-\lfloor\hat{\mathbf{p}}_{L_j}\times\rfloor\mathbf{C}^T(\hat{\mathbf{q}}_{R_0})$ to the first block column, for all $j = 1, \dots n$

$$\sim \left[\begin{array}{ccc|ccccc} \lfloor(-\hat{\mathbf{p}}_{R_0} - \hat{\mathbf{v}}_{R_0}({}^0_0\Delta t) - \frac{\mathbf{g}}{2}({}^0_0\Delta t)^2)\times\rfloor \mathbf{C}^T(\hat{\mathbf{q}}_{R_0}) & -\mathbf{I}_3 & -{}^0_0\Delta t\mathbf{I}_3 & \mathbf{I}_3 & \mathbf{0}_{3\times3} & \dots & \dots & \mathbf{0}_{3\times3} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \lfloor(-\hat{\mathbf{p}}_{R_0} - \hat{\mathbf{v}}_{R_0}({}^i_0\Delta t) - \frac{\mathbf{g}}{2}({}^i_0\Delta t)^2)\times\rfloor \mathbf{C}^T(\hat{\mathbf{q}}_{R_0}) & -\mathbf{I}_3 & -{}^i_0\Delta t\mathbf{I}_3 & \mathbf{0}_{3\times3} & \dots & \mathbf{I}_3 & \dots & \mathbf{0}_{3\times3} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \lfloor(-\hat{\mathbf{p}}_{R_0} - \hat{\mathbf{v}}_{R_0}({}^{k'}_0\Delta t) - \frac{\mathbf{g}}{2}({}^{k'}_0\Delta t)^2)\times\rfloor \mathbf{C}^T(\hat{\mathbf{q}}_{R_0}) & -\mathbf{I}_3 & -{}^{k'}_0\Delta t\mathbf{I}_3 & \mathbf{0}_{3\times3} & \dots & \dots & \dots & \mathbf{I}_3 \end{array}\right]$$

Add the second block column multiplied by $-\lfloor\hat{\mathbf{p}}_{R_0}\times\rfloor\mathbf{C}^T(\hat{\mathbf{q}}_{R_0})$ to the first block column

$$\sim \left[\begin{array}{ccc|ccccc} \lfloor(-\hat{\mathbf{v}}_{R_0}({}^0_0\Delta t) - \frac{\mathbf{g}}{2}({}^0_0\Delta t)^2)\times\rfloor \mathbf{C}^T(\hat{\mathbf{q}}_{R_0}) & -\mathbf{I}_3 & -{}^0_0\Delta t\mathbf{I}_3 & \mathbf{I}_3 & \mathbf{0}_{3\times3} & \dots & \dots & \mathbf{0}_{3\times3} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \lfloor(-\hat{\mathbf{v}}_{R_0}({}^i_0\Delta t) - \frac{\mathbf{g}}{2}({}^i_0\Delta t)^2)\times\rfloor \mathbf{C}^T(\hat{\mathbf{q}}_{R_0}) & -\mathbf{I}_3 & -{}^i_0\Delta t\mathbf{I}_3 & \mathbf{0}_{3\times3} & \dots & \mathbf{I}_3 & \dots & \mathbf{0}_{3\times3} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \lfloor(-\hat{\mathbf{v}}_{R_0}({}^{k'}_0\Delta t) - \frac{\mathbf{g}}{2}({}^{k'}_0\Delta t)^2)\times\rfloor \mathbf{C}^T(\hat{\mathbf{q}}_{R_0}) & -\mathbf{I}_3 & -{}^{k'}_0\Delta t\mathbf{I}_3 & \mathbf{0}_{3\times3} & \dots & \dots & \dots & \mathbf{I}_3 \end{array}\right]$$

Add the third block column multiplied by $-\lfloor\hat{\mathbf{v}}_{R_0}\times\rfloor\mathbf{C}^T(\hat{\mathbf{q}}_{R_0})$ to the first block column

$$\sim \left[\begin{array}{ccc|ccccc} \lfloor(-\frac{\mathbf{g}}{2}({}^0_0\Delta t)^2)\times\rfloor \mathbf{C}^T(\hat{\mathbf{q}}_{R_0}) & -\mathbf{I}_3 & -{}^0_0\Delta t\mathbf{I}_3 & \mathbf{I}_3 & \mathbf{0}_{3\times3} & \dots & \dots & \mathbf{0}_{3\times3} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \lfloor(-\frac{\mathbf{g}}{2}({}^i_0\Delta t)^2)\times\rfloor \mathbf{C}^T(\hat{\mathbf{q}}_{R_0}) & -\mathbf{I}_3 & -{}^i_0\Delta t\mathbf{I}_3 & \mathbf{0}_{3\times3} & \dots & \mathbf{I}_3 & \dots & \mathbf{0}_{3\times3} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \lfloor(-\frac{\mathbf{g}}{2}({}^{k'}_0\Delta t)^2)\times\rfloor \mathbf{C}^T(\hat{\mathbf{q}}_{R_0}) & -\mathbf{I}_3 & -{}^{k'}_0\Delta t\mathbf{I}_3 & \mathbf{0}_{3\times3} & \dots & \dots & \dots & \mathbf{I}_3 \end{array}\right]$$

Add the block columns for $\mathbf{l}_j$ to the second block column, for all $j = 1, \dots n$

$$\sim \left[\begin{array}{ccc|ccccc} \lfloor(-\frac{\mathbf{g}}{2}({}^0_0\Delta t)^2)\times\rfloor \mathbf{C}^T(\hat{\mathbf{q}}_{R_0}) & \mathbf{0}_{3\times3} & -{}^0_0\Delta t\mathbf{I}_3 & \mathbf{I}_3 & \mathbf{0}_{3\times3} & \dots & \dots & \mathbf{0}_{3\times3} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \lfloor(-\frac{\mathbf{g}}{2}({}^i_0\Delta t)^2)\times\rfloor \mathbf{C}^T(\hat{\mathbf{q}}_{R_0}) & \mathbf{0}_{3\times3} & -{}^i_0\Delta t\mathbf{I}_3 & \mathbf{0}_{3\times3} & \dots & \mathbf{I}_3 & \dots & \mathbf{0}_{3\times3} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \lfloor(-\frac{\mathbf{g}}{2}({}^{k'}_0\Delta t)^2)\times\rfloor \mathbf{C}^T(\hat{\mathbf{q}}_{R_0}) & \mathbf{0}_{3\times3} & -{}^{k'}_0\Delta t\mathbf{I}_3 & \mathbf{0}_{3\times3} & \dots & \dots & \dots & \mathbf{I}_3 \end{array}\right]$$

Multiply the first block column by $-2\mathbf{C}(\hat{\mathbf{q}}_{R_0})$

$$\sim \left[\begin{array}{ccc|ccccc} ({}^0_0\Delta t)^2\lfloor\mathbf{g}\times\rfloor & \mathbf{0}_{3\times3} & -{}^0_0\Delta t\mathbf{I}_3 & \mathbf{I}_3 & \mathbf{0}_{3\times3} & \dots & \dots & \mathbf{0}_{3\times3} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ ({}^i_0\Delta t)^2\lfloor\mathbf{g}\times\rfloor & \mathbf{0}_{3\times3} & -{}^i_0\Delta t\mathbf{I}_3 & \mathbf{0}_{3\times3} & \dots & \mathbf{I}_3 & \dots & \mathbf{0}_{3\times3} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ ({}^{k'}_0\Delta t)^2\lfloor\mathbf{g}\times\rfloor & \mathbf{0}_{3\times3} & -{}^{k'}_0\Delta t\mathbf{I}_3 & \mathbf{0}_{3\times3} & \dots & \dots & \dots & \mathbf{I}_3 \end{array}\right] \tag{140}$$

We now define the orthonormal vectors $\mathbf{g}_{p_1}$ and $\mathbf{g}_{p_2}$ that span the plane perpendicular to the gravity vector, $\mathbf{g}$. These vectors have the following properties:

$$\begin{aligned} ||\mathbf{g}_{p_1}||_2 = ||\mathbf{g}_{p_2}||_2 &= 1 \\ \mathbf{g}_{p_1}^T \mathbf{g}_{p_2} = \mathbf{g}_{p_1}^T \mathbf{g} = \mathbf{g}_{p_2}^T \mathbf{g} &= 0 \\ \mathbf{g} \times \mathbf{g}_{p_1}^T = \lfloor \mathbf{g} \times \rfloor \mathbf{g}_{p_1}^T &= -||\mathbf{g}||_2 \mathbf{g}_{p_2} \\ \mathbf{g} \times \mathbf{g}_{p_2}^T = \lfloor \mathbf{g} \times \rfloor \mathbf{g}_{p_2}^T &= ||\mathbf{g}||_2 \mathbf{g}_{p_1} \end{aligned}$$

Moreover, since the vectors $\mathbf{g}_{p_1}$, $\mathbf{g}_{p_2}$, and $\mathbf{g}$ are linearly independent, the matrix

$$\mathbf{F} = \begin{bmatrix} \frac{1}{||\mathbf{g}||_2} \mathbf{g}_{p_2} & \frac{-1}{||\mathbf{g}||_2} \mathbf{g}_{p_1} & \mathbf{g} \end{bmatrix} \tag{141}$$

is nonsingular, and has the property:

$$\lfloor \mathbf{g} \times \rfloor \mathbf{F} = \begin{bmatrix} \mathbf{g}_{p_1} & \mathbf{g}_{p_2} & \mathbf{0}_{3\times 1} \end{bmatrix}$$

Since the matrix $\mathbf{F}$ is nonsingular, we can multiply the first block column in (140) by $\mathbf{F}$ to obtain:

$$\mathbf{K} \sim \left[\begin{array}{ccccc|ccccc} ({}_0^0\Delta t)^2 \begin{bmatrix} \mathbf{g}_{p_1} & \mathbf{g}_{p_2} & \mathbf{0}_{3\times 1} \end{bmatrix} & \mathbf{0}_{3\times 3} & -{}_0^0\Delta t \mathbf{I}_3 & \mathbf{I}_3 & \mathbf{0}_{3\times 3} & \dots & \dots & \mathbf{0}_{3\times 3} \\ \vdots & & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ ({}_0^i\Delta t)^2 \begin{bmatrix} \mathbf{g}_{p_1} & \mathbf{g}_{p_2} & \mathbf{0}_{3\times 1} \end{bmatrix} & \mathbf{0}_{3\times 3} & -{}_0^i\Delta t \mathbf{I}_3 & \mathbf{0}_{3\times 3} & \dots & \mathbf{I}_3 & \dots & \mathbf{0}_{3\times 3} \\ \vdots & & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ ({}_0^{k'}\Delta t)^2 \begin{bmatrix} \mathbf{g}_{p_1} & \mathbf{g}_{p_2} & \mathbf{0}_{3\times 1} \end{bmatrix} & \mathbf{0}_{3\times 3} & -{}_0^{k'}\Delta t \mathbf{I}_3 & \mathbf{0}_{3\times 3} & \dots & \dots & \dots & \mathbf{I}_3 \end{array}\right]$$

Omit zero columns

$$\sim \left[\begin{array}{cc|ccccc} ({}_0^0\Delta t)^2 \mathbf{G} & -{}_0^0\Delta t \mathbf{I}_3 & \mathbf{I}_3 & \mathbf{0}_{3\times 3} & \dots & \dots & \mathbf{0}_{3\times 3} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ ({}_0^i\Delta t)^2 \mathbf{G} & -{}_0^i\Delta t \mathbf{I}_3 & \mathbf{0}_{3\times 3} & \dots & \mathbf{I}_3 & \dots & \mathbf{0}_{3\times 3} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ ({}_0^{k'}\Delta t)^2 \mathbf{G} & -{}_0^{k'}\Delta t \mathbf{I}_3 & \mathbf{0}_{3\times 3} & \dots & \dots & \dots & \mathbf{I}_3 \end{array}\right] = \mathbf{K}_{eq} \tag{142}$$

where $\mathbf{G}$ is a matrix with orthonormal columns:

$$\mathbf{G} = \begin{bmatrix} \mathbf{g}_{p_1} & \mathbf{g}_{p_2} \end{bmatrix}$$

The matrix $\mathbf{K}_{eq}$ in (142) has $3n+5$ columns, therefore its rank (and the rank of the matrix $\mathbf{K}$) is at most $3n+5$. We now show that if there exists at least one landmark that is observed at least 3 times, the rank of the matrix is exactly $3n+5$. To this end, we'll show that the condition $\mathbf{K}_{eq}\mathbf{a} = \mathbf{0}$ is satisfied only when $\mathbf{a} = \mathbf{0}$. We introduce the partitioning for the vector $\mathbf{a}$ given by:

$$\mathbf{a} = \begin{bmatrix} \mathbf{a}_q \\ \mathbf{a}_v \\ \mathbf{a}_1 \\ \mathbf{a}_2 \\ \vdots \\ \mathbf{a}_n \end{bmatrix} \tag{143}$$

where $\mathbf{a}_q$ is a 2×1 vector and $\mathbf{a}_v$, $\mathbf{a}_i$, $i = 1 \dots n$ are 3×1 vectors. With this notation, the condition $\mathbf{K}_{eq}\mathbf{a} = \mathbf{0}$ is equivalent to :

$$({}_0^i\Delta t)^2 \mathbf{G}\mathbf{a}_q + {}_0^i\Delta t \mathbf{a}_v + \mathbf{a}_j = \mathbf{0}, \quad \forall (i,j) \in \mathcal{S} \tag{144}$$

If one landmark is seen at least three times, then for this landmark we obtain the equations:

$$({}_0^a\Delta t)^2 \mathbf{G}\mathbf{a}_q + {}_0^a\Delta t \mathbf{a}_v + \mathbf{a}_{j*} = \mathbf{0}$$

$$({}^b_0\Delta t)^2\mathbf{G}\mathbf{a}_q + {}^b_0\Delta t\mathbf{a}_v + \mathbf{a}_{j*} = \mathbf{0}$$
$$({}^c_0\Delta t)^2\mathbf{G}\mathbf{a}_q + {}^c_0\Delta t\mathbf{a}_v + \mathbf{a}_{j*} = \mathbf{0}$$

It can be easily verified that the only solution to these equations is $\mathbf{a}_q = \mathbf{0}_{2\times 1}$, $\mathbf{a}_v = \mathbf{a}_{j*} = \mathbf{0}_{3\times 1}$. Therefore, if at least one landmark is seen at least three times, both $\mathbf{a}_q$ and $\mathbf{a}_v$ must equal zero, and in turn, from (144) we conclude that $\mathbf{a}_j = \mathbf{0}_{3\times 1}$, for $i = 1 \dots n$. Thus the condition $\mathbf{K}_{eq}\mathbf{a} = \mathbf{0}$ can only be satisfied when $\mathbf{a} = \mathbf{0}$, and thus

$$\text{rank}(\mathbf{K}_{eq}) = \text{rank}(\mathbf{K}) = 3n + 5$$

C.2 Proof of Lemma 7

Relative-position measurements When the robot-to-landmark measurements are direct measurements of the relative position of the landmark with respect to the robot, $\mathbf{z}_{ij} = \Delta\mathbf{p}^{ij} + \mathbf{n}_{ij}$, then the Jacobian $\nabla\mathbf{h}_{ij}$ is simply the identity matrix, and $\mathbf{H}_{L_{ij}} = \mathbf{C}(\mathbf{q}_{R_i})$. Therefore, the matrix $\mathbf{D}(k')$ is a block diagonal matrix with rotation matrices as its diagonal elements, and it is invertible. As a result, $\mathcal{N}(\mathbf{D}(k')) = \emptyset$, which shows that

$$\dim(\mathcal{N}(\mathbf{D}(k'))\bigcap\mathcal{R}(\mathbf{K}(k'))) = 0$$

which is the desired result.

Camera measurements In Appendix C.1, we showed that by applying elementary column operations on the matrix $\mathbf{K}$, we can obtain the matrix $\mathbf{K}_{eq}$ which has $3n + 5$ linearly independent columns. Therefore, the range of the matrix $\mathbf{K}$ is the same as the range of the matrix $\mathbf{K}_{eq}$, and we can write:

$$\mathcal{N}(\mathbf{D})\bigcap\mathcal{R}(\mathbf{K}) = \emptyset \quad \Leftrightarrow \quad \mathcal{N}(\mathbf{D})\bigcap\mathcal{R}(\mathbf{K}_{eq}) = \emptyset \quad \Leftrightarrow \quad \mathbf{D}\mathbf{K}_{eq}\mathbf{a} = \mathbf{0} \text{ only when } \mathbf{a} = \mathbf{0} \tag{145}$$

To prove the above result, we need to examine the structure of the matrix $\mathbf{D}$ appearing in the above expression. When a perspective camera is used for the robot-to-landmark measurements, then

$$\mathbf{h}(\Delta\mathbf{p}^{ij}) = \begin{bmatrix} \frac{{}^c\Delta p_1^{ij}}{{}^c\Delta p_3^{ij}} \\ \frac{{}^c\Delta p_2^{ij}}{{}^c\Delta p_3^{ij}} \end{bmatrix} \qquad \text{with} \qquad \begin{bmatrix} {}^c\Delta p_1^{ij} \\ {}^c\Delta p_2^{ij} \\ {}^c\Delta p_3^{ij} \end{bmatrix} = {}^c_b\mathbf{C}\begin{bmatrix} \Delta p_1^{ij} \\ \Delta p_2^{ij} \\ \Delta p_3^{ij} \end{bmatrix} \tag{146}$$

In the above expression, ${}^c\Delta\mathbf{p}^{ij} = {}^c_b\mathbf{C}\Delta\mathbf{p}^{ij}$ is the feature position expressed in the camera frame, and ${}^c_b\mathbf{C}$ is the 3×3 rotation matrix between the IMU and camera frames (to keep the derivations simpler, we are assuming that the distance between the camera and IMU frames is negligible, compared to the difference between the camera and feature, and can thus be omitted). The Jacobian of $\mathbf{h}$ with respect to $\Delta\mathbf{p}^{ij}$ is:

$$\nabla\mathbf{h}_{ij} = \frac{1}{({}^c\Delta p_3^{ij})^2}\begin{bmatrix} {}^c\Delta p_3^{ij} & 0 & -{}^c\Delta p_1^{ij} \\ 0 & {}^c\Delta p_3^{ij} & -{}^c\Delta p_2^{ij} \end{bmatrix}{}^c_b\mathbf{C} \tag{147}$$

and thus

$$\mathbf{H}_{L_{ij}} = \frac{1}{({}^c\Delta p_3^{ij})^2}\begin{bmatrix} {}^c\Delta p_3^{ij} & 0 & -{}^c\Delta p_1^{ij} \\ 0 & {}^c\Delta p_3^{ij} & -{}^c\Delta p_2^{ij} \end{bmatrix}{}^c_b\mathbf{C}\mathbf{C}(\hat{\mathbf{q}}_{R_i})$$

Using the same partitioning for the vector $\mathbf{a}$ as in (143), we can write the condition $\mathbf{D}\mathbf{K}_{eq}\mathbf{a} = \mathbf{0}$ as:

$$\mathbf{Diag}(\mathbf{H}_{L_{ij}})\left[\begin{array}{cc|ccccc} ({}^0_0\Delta t)^2\mathbf{G} & {}^0_0\Delta t\mathbf{I}_3 & \mathbf{I}_3 & \mathbf{0}_{3\times 3} & \dots & \dots & \mathbf{0}_{3\times 3} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ ({}^i_0\Delta t)^2\mathbf{G} & {}^i_0\Delta t\mathbf{I}_3 & \mathbf{0}_{3\times 3} & \dots & \mathbf{I}_3 & \dots & \mathbf{0}_{3\times 3} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ ({}^{k'}_0\Delta t)^2\mathbf{G} & {}^{k'}_0\Delta t\mathbf{I}_3 & \mathbf{0}_{3\times 3} & \dots & \dots & \dots & \mathbf{I}_3 \end{array}\right]\begin{bmatrix} \mathbf{a}_q \\ \mathbf{a}_v \\ \mathbf{a}_1 \\ \mathbf{a}_2 \\ \vdots \\ \mathbf{a}_n \end{bmatrix} = \mathbf{0}$$
$$\Leftrightarrow \mathbf{H}_{L_{ij}}\left(({}^i_0\Delta t)^2\mathbf{G}\mathbf{a}_q + {}^i_0\Delta t\mathbf{a}_v + \mathbf{a}_j\right) = \mathbf{0}, \quad \forall(i,j) \in \mathcal{S}$$

$$\Leftrightarrow \begin{bmatrix} {}^c\Delta p_3^{ij} & 0 & -{}^c\Delta p_1^{ij} \\ 0 & {}^c\Delta p_3^{ij} & -{}^c\Delta p_2^{ij} \end{bmatrix} {}^c_b\mathbf{C}\mathbf{C}(\hat{\mathbf{q}}_{R_i}) \left(({}^i_0\Delta t)^2 \mathbf{G}\mathbf{a}_q + {}^i_0\Delta t \mathbf{a}_v + \mathbf{a}_j \right) = \mathbf{0}, \quad \forall (i,j) \in \mathcal{S} \tag{148}$$

$$\Leftrightarrow {}^c_b\mathbf{C}\mathbf{C}(\hat{\mathbf{q}}_{R_i}) \left(({}^i_0\Delta t)^2 \mathbf{G}\mathbf{a}_q + {}^i_0\Delta t \mathbf{a}_v + \mathbf{a}_j \right) = c_{ij} {}^c\Delta\mathbf{p}^{ij}, \quad \forall (i,j) \in \mathcal{S} \tag{149}$$

where c_{ij} are scalars. The last equivalence follows from the fact that the matrix $\begin{bmatrix} {}^c\Delta p_3^{ij} & 0 & -{}^c\Delta p_1^{ij} \\ 0 & {}^c\Delta p_3^{ij} & -{}^c\Delta p_2^{ij} \end{bmatrix}$ has a nullspace of dimension one, spanned by the vector ${}^c\Delta\mathbf{p}^{ij}$. For the condition (148) to hold, the vector multiplied by this matrix must lie in its nullspace, which leads to (149). Substituting ${}^c\Delta\mathbf{p}^{ij} = {}^c_b\mathbf{C}\Delta\mathbf{p}^{ij} = {}^c_b\mathbf{C}\mathbf{C}(\hat{\mathbf{q}}_{R_i}) \left(\hat{\mathbf{p}}_{L_j} - \hat{\mathbf{p}}_{R_i} \right)$, we obtain:

$$({}^i_0\Delta t)^2 \mathbf{G}\mathbf{a}_q + {}^i_0\Delta t \mathbf{a}_v + \mathbf{a}_j = c_{ij} \left(\hat{\mathbf{p}}_{L_j} - \hat{\mathbf{p}}_{R_i} \right), \quad \forall (i,j) \in \mathcal{S} \tag{150}$$

Our goal is to examine whether the above conditions can be satisfied for any $\mathbf{a}$ different than zero. To interpret the above condition, note that if the robot trajectory can be described by an equation of the form:

$$\hat{\mathbf{p}}_{R_i} = \hat{\mathbf{p}}_{R_0} + {}^i_0\Delta t \mathbf{v} + ({}^i_0\Delta t)^2 \mathbf{G}\mathbf{a}_{\text{const}} \tag{151}$$

then we can choose $\mathbf{c}_{ij} = 1$, $\mathbf{a}_j = \hat{\mathbf{p}}_{L_j} - \hat{\mathbf{p}}_{R_0}$, $\mathbf{a}_v = -\mathbf{v}$, and $\mathbf{a}_q = -\mathbf{a}_{\text{cosnt}}$, and the condition (150) will be satisfied. In other words, if the robot moves with a constant acceleration, and the acceleration vector is in the plane perpendicular to gravity (the plane spanned by the columns of $\mathbf{G}$), then the condition (150) will be satisfied for $\mathbf{a} \neq 0$. However, if we do not restrict the robot's motion to have any special properties, then in general the condition (150) can only be satisfied for $\mathbf{a} = \mathbf{0}$. Specifically, we prove the following result:

Lemma 10. *If at least two landmarks are observed from at least three different robot poses, and the robot's acceleration is not restricted to the horizontal plane, then Eq.* (150) *implies that* $\mathbf{a} = \mathbf{0}$.

Proof. See Appendix C.3. □

Therefore, if during any point in the robot's trajectory, the conditions of the above lemma are met (two landmarks are observed from three different robot poses, and the robot's trajectory is general), we conclude that $\mathbf{D}\mathbf{K}_{eq}\mathbf{a} = \mathbf{0}$ only when $\mathbf{a} = \mathbf{0}$, which is the desired result.

C.3 Proof of Lemma 10

For two landmarks and three robot poses, Eq. (150) yields the following system of equations:

$$\begin{aligned}
\mathbf{a}_1 + {}^1_0\Delta t\, \mathbf{a}_v + +{}^1_0\Delta t^2 \mathbf{G}\mathbf{a}_q &= c_{11}(\hat{\mathbf{p}}_{L_1} - \hat{\mathbf{p}}_{R_1}) \\
\mathbf{a}_1 + {}^2_0\Delta t\, \mathbf{a}_v + +{}^2_0\Delta t^2 \mathbf{G}\mathbf{a}_q &= c_{21}(\hat{\mathbf{p}}_{L_1} - \hat{\mathbf{p}}_{R_2}) \\
\mathbf{a}_1 + {}^3_0\Delta t\, \mathbf{a}_v + +{}^3_0\Delta t^2 \mathbf{G}\mathbf{a}_q &= c_{31}(\hat{\mathbf{p}}_{L_1} - \hat{\mathbf{p}}_{R_3}) \\
\mathbf{a}_2 + {}^1_0\Delta t\, \mathbf{a}_v + +{}^1_0\Delta t^2 \mathbf{G}\mathbf{a}_q &= c_{12}(\hat{\mathbf{p}}_{L_2} - \hat{\mathbf{p}}_{R_1}) \\
\mathbf{a}_2 + {}^2_0\Delta t\, \mathbf{a}_v + +{}^2_0\Delta t^2 \mathbf{G}\mathbf{a}_q &= c_{22}(\hat{\mathbf{p}}_{L_2} - \hat{\mathbf{p}}_{R_2}) \\
\mathbf{a}_2 + {}^3_0\Delta t\, \mathbf{a}_v + +{}^3_0\Delta t^2 \mathbf{G}\mathbf{a}_q &= c_{32}(\hat{\mathbf{p}}_{L_2} - \hat{\mathbf{p}}_{R_3})
\end{aligned}$$

Note that for simplicity we use the indices 1,2,3 for the three robot poses and 1,2 for the landmarks. This is only for notational clarity, and does not imply that the robot poses are the first three ones, or that the landmarks are the first two ones in the state vector. The above equations can be re-arranged as:

$$\underbrace{\begin{bmatrix} \mathbf{p}_{11} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{I}_3 & \mathbf{0}_{3\times3} & {}^{1}_{0}\Delta t\,\mathbf{I}_3 & \mathbf{G}\,{}^{1}_{0}\Delta t^{\,2} \\ \mathbf{0}_{3\times1} & \mathbf{p}_{12} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times3} & \mathbf{I}_3 & {}^{1}_{0}\Delta t\,\mathbf{I}_3 & \mathbf{G}\,{}^{1}_{0}\Delta t^{\,2} \\ \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{p}_{21} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{I}_3 & \mathbf{0}_{3\times3} & {}^{2}_{0}\Delta t\,\mathbf{I}_3 & \mathbf{G}\,{}^{2}_{0}\Delta t^{\,2} \\ \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{p}_{22} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times3} & \mathbf{I}_3 & {}^{2}_{0}\Delta t\,\mathbf{I}_3 & \mathbf{G}\,{}^{2}_{0}\Delta t^{\,2} \\ \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{p}_{31} & \mathbf{0}_{3\times1} & \mathbf{I}_3 & \mathbf{0}_{3\times3} & {}^{3}_{0}\Delta t\,\mathbf{I}_3 & \mathbf{G}\,{}^{3}_{0}\Delta t^{\,2} \\ \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{p}_{32} & \mathbf{0}_{3\times3} & \mathbf{I}_3 & {}^{3}_{0}\Delta t\,\mathbf{I}_3 & \mathbf{G}\,{}^{3}_{0}\Delta t^{\,2} \end{bmatrix}}_{\mathbf{\Gamma}} \underbrace{\begin{bmatrix} c_{11} \\ c_{12} \\ c_{21} \\ c_{22} \\ c_{31} \\ c_{32} \\ \mathbf{a}_1 \\ \mathbf{a}_2 \\ \mathbf{a}_v \\ \mathbf{a}_q \end{bmatrix}}_{\mathbf{y}} = \mathbf{0} \tag{152}$$

where the vector $\mathbf{y}$ contains all the unknowns in the equations, and we have defined:

$$\mathbf{p}_{ij} = \hat{\mathbf{p}}_{R_i} - \hat{\mathbf{p}}_{L_j},\ i \in \{1,2,3\}, j \in \{1,2\} \tag{153}$$

We will identify the conditions under which the homogeneous equation $\mathbf{\Gamma y} = \mathbf{0}$ has a non-zero solution by analyzing the rank of the 18×17 matrix $\mathbf{\Gamma}$. We perform the following elementary row and column operations on $\mathbf{\Gamma}$:

$$\mathbf{\Gamma} = \begin{bmatrix} \mathbf{p}_{11} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{I}_3 & \mathbf{0}_{3\times3} & {}^{1}_{0}\Delta t\,\mathbf{I}_3 & \mathbf{G}\,{}^{1}_{0}\Delta t^{\,2} \\ \mathbf{0}_{3\times1} & \mathbf{p}_{12} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times3} & \mathbf{I}_3 & {}^{1}_{0}\Delta t\,\mathbf{I}_3 & \mathbf{G}\,{}^{1}_{0}\Delta t^{\,2} \\ \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{p}_{21} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{I}_3 & \mathbf{0}_{3\times3} & {}^{2}_{0}\Delta t\,\mathbf{I}_3 & \mathbf{G}\,{}^{2}_{0}\Delta t^{\,2} \\ \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{p}_{22} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times3} & \mathbf{I}_3 & {}^{2}_{0}\Delta t\,\mathbf{I}_3 & \mathbf{G}\,{}^{2}_{0}\Delta t^{\,2} \\ \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{p}_{31} & \mathbf{0}_{3\times1} & \mathbf{I}_3 & \mathbf{0}_{3\times3} & {}^{3}_{0}\Delta t\,\mathbf{I}_3 & \mathbf{G}\,{}^{3}_{0}\Delta t^{\,2} \\ \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{p}_{32} & \mathbf{0}_{3\times3} & \mathbf{I}_3 & {}^{3}_{0}\Delta t\,\mathbf{I}_3 & \mathbf{G}\,{}^{3}_{0}\Delta t^{\,2} \end{bmatrix} \tag{154}$$

Add the block column for $\mathbf{a}_1$ to the block column for $\mathbf{a}_2$

$$\sim \begin{bmatrix} \mathbf{p}_{11} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{I}_3 & \mathbf{I}_3 & {}^{1}_{0}\Delta t\,\mathbf{I}_3 & \mathbf{G}\,{}^{1}_{0}\Delta t^{\,2} \\ \mathbf{0}_{3\times1} & \mathbf{p}_{12} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times3} & \mathbf{I}_3 & {}^{1}_{0}\Delta t\,\mathbf{I}_3 & \mathbf{G}\,{}^{1}_{0}\Delta t^{\,2} \\ \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{p}_{21} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{I}_3 & \mathbf{I}_3 & {}^{2}_{0}\Delta t\,\mathbf{I}_3 & \mathbf{G}\,{}^{2}_{0}\Delta t^{\,2} \\ \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{p}_{22} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times3} & \mathbf{I}_3 & {}^{2}_{0}\Delta t\,\mathbf{I}_3 & \mathbf{G}\,{}^{2}_{0}\Delta t^{\,2} \\ \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{p}_{31} & \mathbf{0}_{3\times1} & \mathbf{I}_3 & \mathbf{I}_3 & {}^{3}_{0}\Delta t\,\mathbf{I}_3 & \mathbf{G}\,{}^{3}_{0}\Delta t^{\,2} \\ \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{p}_{32} & \mathbf{0}_{3\times3} & \mathbf{I}_3 & {}^{3}_{0}\Delta t\,\mathbf{I}_3 & \mathbf{G}\,{}^{3}_{0}\Delta t^{\,2} \end{bmatrix} \tag{155}$$

Subtract the second block row from the first block row

$$\sim \begin{bmatrix} \mathbf{p}_{11} & -\mathbf{p}_{12} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{I}_3 & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times2} \\ \mathbf{0}_{3\times1} & \mathbf{p}_{12} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times3} & \mathbf{I}_3 & {}^{1}_{0}\Delta t\,\mathbf{I}_3 & \mathbf{G}\,{}^{1}_{0}\Delta t^{\,2} \\ \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{p}_{21} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{I}_3 & \mathbf{I}_3 & {}^{2}_{0}\Delta t\,\mathbf{I}_3 & \mathbf{G}\,{}^{2}_{0}\Delta t^{\,2} \\ \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{p}_{22} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times3} & \mathbf{I}_3 & {}^{2}_{0}\Delta t\,\mathbf{I}_3 & \mathbf{G}\,{}^{2}_{0}\Delta t^{\,2} \\ \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{p}_{31} & \mathbf{0}_{3\times1} & \mathbf{I}_3 & \mathbf{I}_3 & {}^{3}_{0}\Delta t\,\mathbf{I}_3 & \mathbf{G}\,{}^{3}_{0}\Delta t^{\,2} \\ \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{p}_{32} & \mathbf{0}_{3\times3} & \mathbf{I}_3 & {}^{3}_{0}\Delta t\,\mathbf{I}_3 & \mathbf{G}\,{}^{3}_{0}\Delta t^{\,2} \end{bmatrix} \tag{156}$$

We now perform Gauss elimination for the first two block rows. We multiply the first block row with the matrix:

$$\mathbf{T}_{12} = \begin{bmatrix} \mathbf{p}_{11}^T \\ \mathbf{p}_{12}^T \\ (\mathbf{p}_{11} \times \mathbf{p}_{12})^T \end{bmatrix}$$

and perform row operations as follows:

$$\mathbf{\Gamma}_{1:2} = \begin{bmatrix} \mathbf{p}_{11} & -\mathbf{p}_{12} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{I}_3 & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times2} \\ \mathbf{0}_{3\times1} & \mathbf{p}_{12} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times3} & \mathbf{I}_3 & {}^{1}_{0}\Delta t\,\mathbf{I}_3 & \mathbf{G}\,{}^{1}_{0}\Delta t^{\,2} \end{bmatrix}$$

Multiply the first block row by $\mathbf{T}_{12}$

$$\sim \begin{bmatrix} \mathbf{p}_{11}^T\mathbf{p}_{11} & -\mathbf{p}_{11}^T\mathbf{p}_{12} & 0 & 0 & 0 & 0 & \mathbf{p}_{11}^T & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times2} \\ \mathbf{p}_{12}^T\mathbf{p}_{11} & -\mathbf{p}_{12}^T\mathbf{p}_{12} & 0 & 0 & 0 & 0 & \mathbf{p}_{12}^T & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times2} \\ 0 & 0 & 0 & 0 & 0 & 0 & (\mathbf{p}_{11}\times\mathbf{p}_{12})^T & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times2} \\ \mathbf{0}_{3\times1} & \mathbf{p}_{12} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times3} & \mathbf{I}_3 & {}_0^1\Delta t\,\mathbf{I}_3 & \mathbf{G}\,{}_0^1\Delta t^{\,2} \end{bmatrix}$$

Add the first row multiplied by $-\dfrac{\mathbf{p}_{12}^T\mathbf{p}_{11}}{\mathbf{p}_{11}^T\mathbf{p}_{11}}$ to the second row, and define $s_1 = \mathbf{p}_{12}^T\mathbf{p}_{12} - \dfrac{(\mathbf{p}_{12}^T\mathbf{p}_{11})^2}{\mathbf{p}_{11}^T\mathbf{p}_{11}}$

$$\sim \begin{bmatrix} \mathbf{p}_{11}^T\mathbf{p}_{11} & -\mathbf{p}_{11}^T\mathbf{p}_{12} & 0 & 0 & 0 & 0 & \mathbf{p}_{11}^T & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times2} \\ 0 & -s_1 & 0 & 0 & 0 & 0 & \mathbf{p}_{12}^T - \mathbf{p}_{11}^T\frac{\mathbf{p}_{12}^T\mathbf{p}_{11}}{\mathbf{p}_{11}^T\mathbf{p}_{11}} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times2} \\ 0 & 0 & 0 & 0 & 0 & 0 & (\mathbf{p}_{11}\times\mathbf{p}_{12})^T & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times2} \\ \mathbf{0}_{3\times1} & \mathbf{p}_{12} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times3} & \mathbf{I}_3 & {}_0^1\Delta t\,\mathbf{I}_3 & \mathbf{G}\,{}_0^1\Delta t^{\,2} \end{bmatrix}$$

Add the second row multiplied by $\dfrac{1}{s_1}\mathbf{p}_{12}$ to the second block row (rows 3-6)

$$\sim \begin{bmatrix} \mathbf{p}_{11}^T\mathbf{p}_{11} & -\mathbf{p}_{11}^T\mathbf{p}_{12} & 0 & 0 & 0 & 0 & \mathbf{p}_{11}^T & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times2} \\ 0 & -s_1 & 0 & 0 & 0 & 0 & \mathbf{p}_{12}^T - \mathbf{p}_{11}^T\frac{\mathbf{p}_{12}^T\mathbf{p}_{11}}{\mathbf{p}_{11}^T\mathbf{p}_{11}} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times2} \\ 0 & 0 & 0 & 0 & 0 & 0 & (\mathbf{p}_{11}\times\mathbf{p}_{12})^T & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times2} \\ \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \frac{1}{s_1}\mathbf{p}_{12}(\mathbf{p}_{12}^T - \mathbf{p}_{11}^T\frac{\mathbf{p}_{12}^T\mathbf{p}_{11}}{\mathbf{p}_{11}^T\mathbf{p}_{11}}) & \mathbf{I}_3 & {}_0^1\Delta t\,\mathbf{I}_3 & \mathbf{G}\,{}_0^1\Delta t^{\,2} \end{bmatrix}$$

$$\sim \begin{bmatrix} \mathbf{p}_{11}^T\mathbf{p}_{11} & -\mathbf{p}_{11}^T\mathbf{p}_{12} & 0 & 0 & 0 & 0 & \mathbf{p}_{11}^T & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times2} \\ 0 & -s_1 & 0 & 0 & 0 & 0 & \mathbf{p}_{12}^T - \mathbf{p}_{11}^T\frac{\mathbf{p}_{12}^T\mathbf{p}_{11}}{\mathbf{p}_{11}^T\mathbf{p}_{11}} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times2} \\ 0 & 0 & 0 & 0 & 0 & 0 & (\mathbf{p}_{11}\times\mathbf{p}_{12})^T & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times2} \\ \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{F}_1 & \mathbf{I}_3 & {}_0^1\Delta t\,\mathbf{I}_3 & \mathbf{G}\,{}_0^1\Delta t^{\,2} \end{bmatrix}$$

where

$$\mathbf{F}_1 = \frac{1}{s_1}\mathbf{p}_{12}\left(\mathbf{p}_{12}^T - \mathbf{p}_{11}^T\frac{\mathbf{p}_{12}^T\mathbf{p}_{11}}{\mathbf{p}_{11}^T\mathbf{p}_{11}}\right) = \frac{1}{s_1}\mathbf{p}_{12}\mathbf{p}_{12}^T\left(\mathbf{I}_3 - \frac{\mathbf{p}_{11}\mathbf{p}_{11}^T}{\mathbf{p}_{11}^T\mathbf{p}_{11}}\right)$$

Continuing similarly for the third-fourth and fifth-sixth block rows, after performing elimination and re-arranging the rows, we obtain:

$$\boldsymbol{\Gamma} \sim \left[\begin{array}{cccccc|cccc} \mathbf{p}_{11}^T\mathbf{p}_{11} & -\mathbf{p}_{11}^T\mathbf{p}_{12} & 0 & 0 & 0 & 0 & \mathbf{p}_{11}^T & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times2} \\ 0 & -s_1 & 0 & 0 & 0 & 0 & \mathbf{p}_{12}^T - \mathbf{p}_{11}^T\frac{\mathbf{p}_{12}^T\mathbf{p}_{11}}{\mathbf{p}_{11}^T\mathbf{p}_{11}} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times2} \\ 0 & 0 & \mathbf{p}_{21}^T\mathbf{p}_{21} & -\mathbf{p}_{21}^T\mathbf{p}_{22} & 0 & 0 & \mathbf{p}_{21}^T & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times2} \\ 0 & 0 & 0 & -s_2 & 0 & 0 & \mathbf{p}_{22}^T - \mathbf{p}_{21}^T\frac{\mathbf{p}_{22}^T\mathbf{p}_{21}}{\mathbf{p}_{21}^T\mathbf{p}_{21}} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times2} \\ 0 & 0 & 0 & 0 & \mathbf{p}_{31}^T\mathbf{p}_{31} & -\mathbf{p}_{31}^T\mathbf{p}_{32} & \mathbf{p}_{31}^T & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times2} \\ 0 & 0 & 0 & 0 & 0 & -s_3 & \mathbf{p}_{32}^T - \mathbf{p}_{31}^T\frac{\mathbf{p}_{32}^T\mathbf{p}_{31}}{\mathbf{p}_{31}^T\mathbf{p}_{31}} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times2} \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & (\mathbf{p}_{11}\times\mathbf{p}_{12})^T & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times2} \\ 0 & 0 & 0 & 0 & 0 & 0 & (\mathbf{p}_{21}\times\mathbf{p}_{22})^T & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times2} \\ 0 & 0 & 0 & 0 & 0 & 0 & (\mathbf{p}_{31}\times\mathbf{p}_{32})^T & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times2} \\ \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{F}_1 & \mathbf{I}_3 & {}_0^1\Delta t\,\mathbf{I}_3 & \mathbf{G}\,{}_0^1\Delta t^{\,2} \\ \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{F}_3 & \mathbf{I}_3 & {}_0^2\Delta t\,\mathbf{I}_3 & \mathbf{G}\,{}_0^2\Delta t^{\,2} \\ \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{F}_2 & \mathbf{I}_3 & {}_0^3\Delta t\,\mathbf{I}_3 & \mathbf{G}\,{}_0^3\Delta t^{\,2} \end{array}\right] \tag{157}$$

where

$$s_2 = \mathbf{p}_{22}^T\mathbf{p}_{22} - \frac{(\mathbf{p}_{22}^T\mathbf{p}_{21})^2}{\mathbf{p}_{21}^T\mathbf{p}_{21}}, \;\; s_3 = \mathbf{p}_{32}^T\mathbf{p}_{32} - \frac{(\mathbf{p}_{32}^T\mathbf{p}_{31})^2}{\mathbf{p}_{31}^T\mathbf{p}_{31}} \tag{158}$$

$$\mathbf{F}_2 = \frac{1}{s_2}\mathbf{p}_{22}\mathbf{p}_{22}^T\left(\mathbf{I}_3 - \frac{\mathbf{p}_{21}\mathbf{p}_{21}^T}{\mathbf{p}_{21}^T\mathbf{p}_{21}}\right), \;\; \mathbf{F}_3 = \frac{1}{s_3}\mathbf{p}_{32}\mathbf{p}_{32}^T\left(\mathbf{I}_3 - \frac{\mathbf{p}_{31}\mathbf{p}_{31}^T}{\mathbf{p}_{31}^T\mathbf{p}_{31}}\right) \tag{159}$$

Note that in the above expressions we have divided by the terms s_i, $i = 1, 2, 3$. To prove that these terms are nonzero, note that

$$s_i = 0 \;\; \Leftrightarrow \;\; \mathbf{p}_{i2}^T\mathbf{p}_{i2}\,\mathbf{p}_{i1}^T\mathbf{p}_{i1} - (\mathbf{p}_{i2}^T\mathbf{p}_{i1})^2 = 0 \;\; \Leftrightarrow \;\; ||\mathbf{p}_{i2}||_2^2||\mathbf{p}_{i1}||_2^2 - (\mathbf{p}_{i2}^T\mathbf{p}_{i1})^2 = 0 \tag{160}$$

By the Cauchy-Schwartz inequality, we know that the above condition is met only when $\mathbf{p}_{i1} = c\mathbf{p}_{i2}$, i.e., only when the vectors $\mathbf{p}_{i1}$ and $\mathbf{p}_{i2}$ are parallel. This is not possible however, since in that case the two landmarks would be projected to the same point on the camera.

From (157) we conclude that

$$\text{rank}(\mathbf{\Gamma}) = 6 + \text{rank}(\mathbf{B}) \tag{161}$$

with

$$\mathbf{B} = \begin{bmatrix} (\mathbf{p}_{11} \times \mathbf{p}_{12})^T & \mathbf{0}_{1\times 3} & \mathbf{0}_{1\times 3} & \mathbf{0}_{1\times 2} \\ (\mathbf{p}_{21} \times \mathbf{p}_{22})^T & \mathbf{0}_{1\times 3} & \mathbf{0}_{1\times 3} & \mathbf{0}_{1\times 2} \\ (\mathbf{p}_{31} \times \mathbf{p}_{32})^T & \mathbf{0}_{1\times 3} & \mathbf{0}_{1\times 3} & \mathbf{0}_{1\times 2} \\ \mathbf{F}_1 & \mathbf{I}_3 & {}_0^1\Delta t\,\mathbf{I}_3 & \mathbf{G}\,{}_0^1\Delta t^{\,2} \\ \mathbf{F}_3 & \mathbf{I}_3 & {}_0^2\Delta t\,\mathbf{I}_3 & \mathbf{G}\,{}_0^2\Delta t^{\,2} \\ \mathbf{F}_2 & \mathbf{I}_3 & {}_0^3\Delta t\,\mathbf{I}_3 & \mathbf{G}\,{}_0^3\Delta t^{\,2} \end{bmatrix} \tag{162}$$

Re-arranging the columns and rows of $\mathbf{B}$, we can write:

$$\mathbf{B} \sim \begin{bmatrix} \mathbf{I}_3 & {}_0^1\Delta t\,\mathbf{I}_3 & \mathbf{G}\,{}_0^1\Delta t^{\,2} & \mathbf{F}_1 \\ \mathbf{I}_3 & {}_0^2\Delta t\,\mathbf{I}_3 & \mathbf{G}\,{}_0^2\Delta t^{\,2} & \mathbf{F}_2 \\ \mathbf{I}_3 & {}_0^3\Delta t\,\mathbf{I}_3 & \mathbf{G}\,{}_0^3\Delta t^{\,2} & \mathbf{F}_3 \\ \mathbf{0}_{1\times 3} & \mathbf{0}_{1\times 3} & \mathbf{0}_{1\times 2} & (\mathbf{p}_{11} \times \mathbf{p}_{12})^T \\ \mathbf{0}_{1\times 3} & \mathbf{0}_{1\times 3} & \mathbf{0}_{1\times 2} & (\mathbf{p}_{21} \times \mathbf{p}_{22})^T \\ \mathbf{0}_{1\times 3} & \mathbf{0}_{1\times 3} & \mathbf{0}_{1\times 2} & (\mathbf{p}_{31} \times \mathbf{p}_{32})^T \end{bmatrix} \tag{163}$$

Subtract the first block row from the second and third block rows

$$\sim \begin{bmatrix} \mathbf{I}_3 & {}_0^1\Delta t\,\mathbf{I}_3 & \mathbf{G}\,{}_0^1\Delta t^{\,2} & \mathbf{F}_1 \\ \mathbf{0}_{3\times 3} & ({}_0^2\Delta t - {}_0^1\Delta t)\mathbf{I}_3 & ({}_0^2\Delta t^{\,2} - {}_0^1\Delta t^{\,2})\mathbf{G} & \mathbf{F}_2 - \mathbf{F}_1 \\ \mathbf{0}_{3\times 3} & ({}_0^3\Delta t - {}_0^1\Delta t)\mathbf{I}_3 & ({}_0^3\Delta t^{\,2} - {}_0^1\Delta t^{\,2})\mathbf{G} & \mathbf{F}_3 - \mathbf{F}_1 \\ \mathbf{0}_{1\times 3} & \mathbf{0}_{1\times 3} & \mathbf{0}_{1\times 2} & (\mathbf{p}_{11} \times \mathbf{p}_{12})^T \\ \mathbf{0}_{1\times 3} & \mathbf{0}_{1\times 3} & \mathbf{0}_{1\times 2} & (\mathbf{p}_{21} \times \mathbf{p}_{22})^T \\ \mathbf{0}_{1\times 3} & \mathbf{0}_{1\times 3} & \mathbf{0}_{1\times 2} & (\mathbf{p}_{31} \times \mathbf{p}_{32})^T \end{bmatrix} \tag{164}$$

Divide the second and third block rows by $({}_0^2\Delta t - {}_0^1\Delta t)$ and $({}_0^3\Delta t - {}_0^1\Delta t)$, respectively

$$\sim \begin{bmatrix} \mathbf{I}_3 & {}_0^1\Delta t\,\mathbf{I}_3 & \mathbf{G}\,{}_0^1\Delta t^{\,2} & \mathbf{F}_1 \\ \mathbf{0}_{3\times 3} & \mathbf{I}_3 & ({}_0^2\Delta t + {}_0^1\Delta t)\mathbf{G} & \frac{1}{{}_0^2\Delta t - {}_0^1\Delta t}(\mathbf{F}_2 - \mathbf{F}_1) \\ \mathbf{0}_{3\times 3} & \mathbf{I}_3 & ({}_0^3\Delta t + {}_0^1\Delta t)\mathbf{G} & \frac{1}{{}_0^3\Delta t - {}_0^1\Delta t}(\mathbf{F}_3 - \mathbf{F}_1) \\ \mathbf{0}_{1\times 3} & \mathbf{0}_{1\times 3} & \mathbf{0}_{1\times 2} & (\mathbf{p}_{11} \times \mathbf{p}_{12})^T \\ \mathbf{0}_{1\times 3} & \mathbf{0}_{1\times 3} & \mathbf{0}_{1\times 2} & (\mathbf{p}_{21} \times \mathbf{p}_{22})^T \\ \mathbf{0}_{1\times 3} & \mathbf{0}_{1\times 3} & \mathbf{0}_{1\times 2} & (\mathbf{p}_{31} \times \mathbf{p}_{32})^T \end{bmatrix} \tag{165}$$

Subtract the second block row from the third block row

$$\sim \begin{bmatrix} \mathbf{I}_3 & {}_0^1\Delta t\,\mathbf{I}_3 & \mathbf{G}\,{}_0^1\Delta t^{\,2} & \mathbf{F}_1 \\ \mathbf{0}_{3\times 3} & \mathbf{I}_3 & ({}_0^2\Delta t + {}_0^1\Delta t)\mathbf{G} & \frac{1}{{}_0^2\Delta t - {}_0^1\Delta t}(\mathbf{F}_2 - \mathbf{F}_1) \\ \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & ({}_0^3\Delta t - {}_0^2\Delta t)\mathbf{G} & \frac{1}{{}_0^3\Delta t - {}_0^1\Delta t}(\mathbf{F}_3 - \mathbf{F}_1) - \frac{1}{{}_0^2\Delta t - {}_0^1\Delta t}(\mathbf{F}_2 - \mathbf{F}_1) \\ \mathbf{0}_{1\times 3} & \mathbf{0}_{1\times 3} & \mathbf{0}_{1\times 2} & (\mathbf{p}_{11} \times \mathbf{p}_{12})^T \\ \mathbf{0}_{1\times 3} & \mathbf{0}_{1\times 3} & \mathbf{0}_{1\times 2} & (\mathbf{p}_{21} \times \mathbf{p}_{22})^T \\ \mathbf{0}_{1\times 3} & \mathbf{0}_{1\times 3} & \mathbf{0}_{1\times 2} & (\mathbf{p}_{31} \times \mathbf{p}_{32})^T \end{bmatrix} \tag{166}$$

Pre-multiply the third block row with the matrix $[\mathbf{G}\ \ \mathbf{g}]^T$

$$\sim \left[\begin{array}{ccc|c} \mathbf{I}_3 & {}_0^1\Delta t\,\mathbf{I}_3 & \mathbf{G}\,{}_0^1\Delta t^{\,2} & \mathbf{F}_1 \\ \mathbf{0}_{3\times3} & \mathbf{I}_3 & ({}_0^2\Delta t + {}_0^1\Delta t)\mathbf{G} & \frac{1}{{}_0^2\Delta t - {}_0^1\Delta t}(\mathbf{F}_2 - \mathbf{F}_1) \\ \mathbf{0}_{2\times3} & \mathbf{0}_{2\times3} & ({}_0^3\Delta t - {}_0^2\Delta t)\mathbf{I}_2 & \mathbf{B}' \\ \hline \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times2} & \frac{1}{{}_0^3\Delta t - {}_0^1\Delta t}\mathbf{g}^T(\mathbf{F}_3 - \mathbf{F}_1) - \frac{1}{{}_0^2\Delta t - {}_0^1\Delta t}\mathbf{g}^T(\mathbf{F}_2 - \mathbf{F}_1) \\ \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times2} & (\mathbf{p}_{11}\times\mathbf{p}_{12})^T \\ \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times2} & (\mathbf{p}_{21}\times\mathbf{p}_{22})^T \\ \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times2} & (\mathbf{p}_{31}\times\mathbf{p}_{32})^T \end{array}\right] \tag{167}$$

where $\mathbf{B}'$ is a 2×3 matrix whose value is not required in this derivation. The top-left submatrix above is full rank, and therefore we can write

$$\mathrm{rank}(\mathbf{B}) = 8 + \mathrm{rank}(\mathbf{\Theta}) \tag{168}$$

where

$$\mathbf{\Theta} = \begin{bmatrix} \mathbf{g}^T\big(({}_0^2\Delta t - {}_0^1\Delta t)(\mathbf{F}_3 - \mathbf{F}_1) - ({}_0^3\Delta t - {}_0^1\Delta t)(\mathbf{F}_2 - \mathbf{F}_1)\big) \\ (\mathbf{p}_{11}\times\mathbf{p}_{12})^T \\ (\mathbf{p}_{21}\times\mathbf{p}_{22})^T \\ (\mathbf{p}_{31}\times\mathbf{p}_{32})^T \end{bmatrix} \tag{169}$$

From (161) and (168) we conclude that the matrix $\mathbf{\Gamma}$ is full rank when $\mathrm{rank}(\mathbf{\Theta}) = 3$. We will therefore examine the conditions under which this is true.

Using Eq. (153) we see that, for $i = \{1,2,3\}$, it is:

$$\begin{aligned} \mathbf{p}_{i1}\times\mathbf{p}_{i2} &= (\hat{\mathbf{p}}_{R_i} - \hat{\mathbf{p}}_{L_1}) \times (\hat{\mathbf{p}}_{R_i} - \hat{\mathbf{p}}_{L_2}) \\ &= \hat{\mathbf{p}}_{L_1}\times\hat{\mathbf{p}}_{L_2} + \hat{\mathbf{p}}_{R_i}\times(\hat{\mathbf{p}}_{L_1} - \hat{\mathbf{p}}_{L_2}) \end{aligned}$$

and using the fact that $\hat{\mathbf{p}}_{L_1}\times\hat{\mathbf{p}}_{L_2} \perp (\hat{\mathbf{p}}_{L_1} - \hat{\mathbf{p}}_{L_2})$ and $\hat{\mathbf{p}}_{R_i}\times(\hat{\mathbf{p}}_{L_1} - \hat{\mathbf{p}}_{L_2}) \perp (\hat{\mathbf{p}}_{L_1} - \hat{\mathbf{p}}_{L_2})$, we obtain

$$(\mathbf{p}_{i1}\times\mathbf{p}_{i2})^T(\hat{\mathbf{p}}_{L_1} - \hat{\mathbf{p}}_{L_2}) = 0 \tag{170}$$

Therefore, the vectors $(\mathbf{p}_{11}\times\mathbf{p}_{12}), (\mathbf{p}_{21}\times\mathbf{p}_{22}), (\mathbf{p}_{31}\times\mathbf{p}_{32})$ lie in a subspace of dimension two (this subspace is the plane with normal vector $(\hat{\mathbf{p}}_{L_1} - \hat{\mathbf{p}}_{L_2})$). More precisely, we have proven that the vectors lie in a subspace of dimension *at most* two. They will lie in a subspace of dimension one only when $\mathbf{p}_{i1}\times\mathbf{p}_{i2} = c_i\mathbf{d}$ for some $\mathbf{d}$, which means that all the robot positions and all the landmarks are in the same plane. Unless this is the case, then the vectors $(\mathbf{p}_{11}\times\mathbf{p}_{12}), (\mathbf{p}_{21}\times\mathbf{p}_{22}), (\mathbf{p}_{31}\times\mathbf{p}_{32})$ span a subspace of dimension two. As a result,

$$\mathrm{rank}\left(\begin{bmatrix}(\mathbf{p}_{11}\times\mathbf{p}_{12})^T \\ (\mathbf{p}_{21}\times\mathbf{p}_{22})^T \\ (\mathbf{p}_{31}\times\mathbf{p}_{32})^T\end{bmatrix}\right) = 2$$

and any vector $\mathbf{w}$ such that

$$\begin{bmatrix}(\mathbf{p}_{11}\times\mathbf{p}_{12})^T \\ (\mathbf{p}_{21}\times\mathbf{p}_{22})^T \\ (\mathbf{p}_{31}\times\mathbf{p}_{32})^T\end{bmatrix}\mathbf{w} = \mathbf{0}$$

must lie in the nullspace of this matrix, which is of dimension one. Since the vector $(\hat{\mathbf{p}}_{L_1} - \hat{\mathbf{p}}_{L_2})$ lies in this nullspace (see (170)), we must have that $\mathbf{w} = c(\hat{\mathbf{p}}_{L_1} - \hat{\mathbf{p}}_{L_2})$ for some constant c.

We have thus shown that if there exists any vector such that $\mathbf{\Theta}\mathbf{w} = \mathbf{0}$, this vector can only be of the form $\mathbf{w} = c(\hat{\mathbf{p}}_{L_1} - \hat{\mathbf{p}}_{L_2})$. For $\mathbf{\Theta}\mathbf{w} = \mathbf{0}$ to hold, from the first row of the matrix $\mathbf{\Theta}$ we obtain the condition:

$$c = 0 \quad \text{or} \quad \mathbf{g}^T\big(({}_0^2\Delta t - {}_0^1\Delta t)(\mathbf{F}_3 - \mathbf{F}_1) - ({}_0^3\Delta t - {}_0^1\Delta t)(\mathbf{F}_2 - \mathbf{F}_1)\big)(\hat{\mathbf{p}}_{L_1} - \hat{\mathbf{p}}_{L_2}) = 0 \tag{171}$$

We now note that

$$\mathbf{F}_1(\hat{\mathbf{p}}_{L_1} - \hat{\mathbf{p}}_{L_2}) = \frac{1}{s_1}\mathbf{p}_{12}\mathbf{p}_{12}^T\left(\mathbf{I}_3 - \frac{\mathbf{p}_{11}\mathbf{p}_{11}^T}{\mathbf{p}_{11}^T\mathbf{p}_{11}}\right)(\hat{\mathbf{p}}_{L_1} - \hat{\mathbf{p}}_{L_2})$$

$$
\begin{aligned}
&= \frac{1}{s_1}\mathbf{p}_{12}\mathbf{p}_{12}^T\left(\mathbf{I}_3 - \frac{\mathbf{p}_{11}\mathbf{p}_{11}^T}{\mathbf{p}_{11}^T\mathbf{p}_{11}}\right)(\mathbf{p}_{12}-\mathbf{p}_{11}) \\
&= \frac{1}{s_1}\mathbf{p}_{12}\mathbf{p}_{12}^T\left(\mathbf{p}_{12}\underbrace{-\mathbf{p}_{11}+\frac{\mathbf{p}_{11}\mathbf{p}_{11}^T}{\mathbf{p}_{11}^T\mathbf{p}_{11}}\mathbf{p}_{11}}_{0}-\frac{\mathbf{p}_{11}\mathbf{p}_{11}^T}{\mathbf{p}_{11}^T\mathbf{p}_{11}}\mathbf{p}_{12}\right) \\
&= \frac{1}{s_1}\mathbf{p}_{12}\underbrace{\left(\mathbf{p}_{12}^T\mathbf{p}_{12}-\mathbf{p}_{12}^T\frac{\mathbf{p}_{11}\mathbf{p}_{11}^T}{\mathbf{p}_{11}^T\mathbf{p}_{11}}\mathbf{p}_{12}\right)}_{s_1} \\
&= \mathbf{p}_{12}
\end{aligned} \tag{172}
$$

Similarly we obtain:

$$
\begin{aligned}
\mathbf{F}_2(\hat{\mathbf{p}}_{L_1}-\hat{\mathbf{p}}_{L_2}) &= \mathbf{p}_{22} \\
\mathbf{F}_3(\hat{\mathbf{p}}_{L_1}-\hat{\mathbf{p}}_{L_2}) &= \mathbf{p}_{32}
\end{aligned}
$$

With these results, (171) becomes:

$$
c=0 \quad \text{or} \quad \mathbf{g}^T\big(({}^2_0\Delta t - {}^1_0\Delta t)(\mathbf{p}_{32}-\mathbf{p}_{12}) - ({}^3_0\Delta t - {}^1_0\Delta t)(\mathbf{p}_{22}-\mathbf{p}_{12})\big)=0 \tag{173}
$$

the second condition above means that the vector $({}^2_0\Delta t - {}^1_0\Delta t)(\mathbf{p}_{32}-\mathbf{p}_{12}) - ({}^3_0\Delta t - {}^1_0\Delta t)(\mathbf{p}_{22}-\mathbf{p}_{12})$ must be perpendicular to $\mathbf{g}$, i.e., that there exist scalars a_1 and a_2 such that

$$
\begin{aligned}
({}^2_0\Delta t - {}^1_0\Delta t)\underbrace{(\mathbf{p}_{32}-\mathbf{p}_{12})}_{\hat{\mathbf{p}}_{R_3}-\hat{\mathbf{p}}_{R_1}} - ({}^3_0\Delta t - {}^1_0\Delta t)\underbrace{(\mathbf{p}_{22}-\mathbf{p}_{12})}_{\hat{\mathbf{p}}_{R_2}-\hat{\mathbf{p}}_{R_1}} &= \mathbf{G}\begin{bmatrix} a_1 \\ a_2 \end{bmatrix} \\
\Leftrightarrow \frac{\hat{\mathbf{p}}_{R_3}-\hat{\mathbf{p}}_{R_1}}{{}^3_0\Delta t - {}^1_0\Delta t} - \frac{\hat{\mathbf{p}}_{R_2}-\hat{\mathbf{p}}_{R_1}}{{}^2_0\Delta t - {}^1_0\Delta t} &= \mathbf{G}\begin{bmatrix} a_1 \\ a_2 \end{bmatrix}\frac{1}{({}^2_0\Delta t - {}^1_0\Delta t)({}^3_0\Delta t - {}^1_0\Delta t)} \\
\Leftrightarrow \underbrace{\frac{\hat{\mathbf{p}}_{R_3}-\hat{\mathbf{p}}_{R_1}}{{}^3_0\Delta t - {}^1_0\Delta t}}_{\mathbf{v}_{31}} - \underbrace{\frac{\hat{\mathbf{p}}_{R_2}-\hat{\mathbf{p}}_{R_1}}{{}^2_0\Delta t - {}^1_0\Delta t}}_{\mathbf{v}_{21}} &= \mathbf{G}\begin{bmatrix} a_1' \\ a_2' \end{bmatrix}
\end{aligned} \tag{174}
$$

The physical interpretation of the last condition is that the velocity change is perpendicular to $\mathbf{g}$, which implies that the robot's acceleration lies in the horizontal plane. Thus, if this is *not* the case, then the last condition is not satisfied, and c must equal zero, which in turn means that $\mathbf{\Theta}\mathbf{w}=\mathbf{0}$ is true only when $\mathbf{w}=\mathbf{0}$, and $\mathrm{rank}(\mathbf{\Theta})=3$. This completes the proof.

C.4 Proof of Lemma 8

For the purposes of this proof, we follow a course analogous to that in Appendix C.1. Specifically, by steps similar to those of Appendix C.1 we obtain

$$
\mathbf{K}_R(k,k') = \begin{bmatrix} \mathbf{H}'_{R_0}(k) \\ \vdots \\ \mathbf{H}'_{R_{m-1}}(k)\mathbf{\Phi}_{R_{m-2}}(k)\ldots\mathbf{\Phi}_{R_0}(k) \\ \mathbf{H}'_{R_m}(k')\mathbf{\Phi}_{R_{m-1}}(k)\ldots\mathbf{\Phi}_{R_0}(k) \\ \vdots \\ \mathbf{H}'_{R_{k'}}(k')\mathbf{\Phi}_{R_{k'-1}}(k')\ldots\mathbf{\Phi}_{R_0}(k) \end{bmatrix} \tag{175}
$$

$$= \begin{bmatrix} \lfloor(\hat{\mathbf{p}}_{L_1}(k) - \hat{\mathbf{p}}_{R_0}(k) - \hat{\mathbf{v}}_{R_0}(k)({}^0_0\Delta t) - \frac{\mathbf{g}}{2}({}^0_0\Delta t)^2)\times\rfloor \mathbf{C}^T(\hat{\mathbf{q}}_{R_0}(k)) & -\mathbf{I}_3 & -{}^0_0\Delta t\mathbf{I}_3 \\ \vdots & \vdots & \vdots \\ \lfloor(\hat{\mathbf{p}}_{L_a}(k) - \hat{\mathbf{p}}_{R_0}(k) - \hat{\mathbf{v}}_{R_0}(k)({}^{m-1}_0\Delta t) - \frac{\mathbf{g}}{2}({}^{m-1}_0\Delta t)^2)\times\rfloor \mathbf{C}^T(\hat{\mathbf{q}}_{R_0}(k)) & -\mathbf{I}_3 & -{}^{m-1}_0\Delta t\mathbf{I}_3 \\ \hline \left(\lfloor(\hat{\mathbf{p}}_{L_a}(k') - \hat{\mathbf{p}}_{R_0}(k) - \hat{\mathbf{v}}_{R_0}(k)({}^m_0\Delta t) - \frac{\mathbf{g}}{2}({}^m_0\Delta t)^2)\times\rfloor + \mathbf{V}_{ma}\right) \mathbf{C}^T(\hat{\mathbf{q}}_{R_0}(k)) & -\mathbf{I}_3 & -{}^m_0\Delta t\mathbf{I}_3 \\ \vdots & \vdots & \vdots \\ \left(\lfloor(\hat{\mathbf{p}}_{L_n}(k') - \hat{\mathbf{p}}_{R_0}(k) - \hat{\mathbf{v}}_{R_0}(k)({}^{k'}_0\Delta t) - \frac{\mathbf{g}}{2}({}^{k'}_0\Delta t)^2)\times\rfloor + \mathbf{V}_{k'n}\right) \mathbf{C}^T(\hat{\mathbf{q}}_{R_0}(k)) & -\mathbf{I}_3 & -{}^{k'}_0\Delta t\mathbf{I}_3 \end{bmatrix} \tag{176}$$

where

$$\mathbf{V}_{ij} = \underbrace{\lfloor\hat{\mathbf{p}}_{L_j}(k')\times\rfloor \left(\mathbf{I}_3 - \mathbf{C}^T(\mathbf{q}_{R_m}(k'))\mathbf{C}(\mathbf{q}_{R_m}(k))\right) - \lfloor\hat{\mathbf{p}}_{R_m}(k')\times\rfloor\mathbf{C}^T(\mathbf{q}_{R_m}(k'))\mathbf{C}(\mathbf{q}_{R_m}(k)) + \lfloor\hat{\mathbf{p}}_{R_m}(k)\times\rfloor}_{\mathbf{\Xi}_j}$$
$$+ \underbrace{\frac{{}^i_m\Delta t^2}{2}\lfloor\mathbf{g}\times\rfloor \left(\mathbf{I}_3 - \mathbf{C}^T(\mathbf{q}_{R_m}(k'))\mathbf{C}(\mathbf{q}_{R_m}(k))\right) + {}^i_m\Delta t \left(\lfloor\hat{\mathbf{v}}_{R_m}(k)\times\rfloor - \lfloor\hat{\mathbf{v}}_{R_m}(k')\times\rfloor\mathbf{C}^T(\mathbf{q}_{R_m}(k'))\mathbf{C}(\mathbf{q}_{R_m}(k))\right)}_{\mathbf{\Lambda}_i}$$

Similarly to Appendix B.3, we here consider the case where the landmark $\mathbf{l}_a$ is observed by both the marginalized state $\mathbf{r}_{m-1}$, and the non-marginalized state $\mathbf{r}_m$. Substituting the above result in the matrix $\mathbf{K}(k,k') = [\mathbf{K}_{R(k,k')} \quad \mathbf{K}_L]$, and applying elementary column operations as in Appendix C.1, we obtain:

$$\mathbf{K}(k,k') \sim \left[\begin{array}{ccc|ccccc} \lfloor(\hat{\mathbf{p}}_{L_1}(k) - \frac{\mathbf{g}}{2}({}^0_0\Delta t)^2)\times\rfloor & \mathbf{0}_{3\times3} & {}^0_0\Delta t\mathbf{I}_3 & \mathbf{I}_3 & \mathbf{0}_{3\times3} & \dots & \dots & \mathbf{0}_{3\times3} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \lfloor(\hat{\mathbf{p}}_{L_a}(k) - \frac{\mathbf{g}}{2}({}^{m-1}_0\Delta t)^2)\times\rfloor & \mathbf{0}_{3\times3} & {}^{m-1}_0\Delta t\mathbf{I}_3 & \mathbf{0}_{3\times3} & \dots & \mathbf{I}_3 & \dots & \mathbf{0}_{3\times3} \\ \hline \lfloor(\hat{\mathbf{p}}_{L_a}(k') - \frac{\mathbf{g}}{2}({}^m_0\Delta t)^2)\times\rfloor + \mathbf{V}_{ma} & \mathbf{0}_{3\times3} & {}^m_0\Delta t\mathbf{I}_3 & \mathbf{0}_{3\times3} & \dots & \mathbf{I}_3 & \dots & \mathbf{0}_{3\times3} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \lfloor(\hat{\mathbf{p}}_{L_n}(k') - \frac{\mathbf{g}}{2}({}^{k'}_0\Delta t)^2)\times\rfloor + \mathbf{V}_{k'n} & \mathbf{0}_{3\times3} & {}^{k'}_0\Delta t\mathbf{I}_3 & \mathbf{0}_{3\times3} & \dots & \dots & \dots & \mathbf{I}_3 \end{array}\right]$$

Omit zero columns

$$\sim \left[\begin{array}{cc|ccccc} \lfloor(\hat{\mathbf{p}}_{L_1}(k) - \frac{\mathbf{g}}{2}({}^0_0\Delta t)^2)\times\rfloor & {}^0_0\Delta t\mathbf{I}_3 & \mathbf{I}_3 & \mathbf{0}_{3\times3} & \dots & \dots & \mathbf{0}_{3\times3} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \lfloor(\hat{\mathbf{p}}_{L_a}(k) - \frac{\mathbf{g}}{2}({}^{m-1}_0\Delta t)^2)\times\rfloor & {}^{m-1}_0\Delta t\mathbf{I}_3 & \mathbf{0}_{3\times3} & \dots & \mathbf{I}_3 & \dots & \mathbf{0}_{3\times3} \\ \hline \lfloor(\hat{\mathbf{p}}_{L_a}(k') - \frac{\mathbf{g}}{2}({}^m_0\Delta t)^2)\times\rfloor + \mathbf{V}_{ma} & {}^m_0\Delta t\mathbf{I}_3 & \mathbf{0}_{3\times3} & \dots & \mathbf{I}_3 & \dots & \mathbf{0}_{3\times3} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \lfloor(\hat{\mathbf{p}}_{L_n}(k') - \frac{\mathbf{g}}{2}({}^{k'}_0\Delta t)^2)\times\rfloor + \mathbf{V}_{k'n} & {}^{k'}_0\Delta t\mathbf{I}_3 & \mathbf{0}_{3\times3} & \dots & \dots & \dots & \mathbf{I}_3 \end{array}\right]$$

$$\sim \left[\begin{array}{cc|ccccc} \lfloor-\frac{\mathbf{g}}{2}({}^0_0\Delta t)^2\times\rfloor & {}^0_0\Delta t\mathbf{I}_3 & \mathbf{I}_3 & \mathbf{0}_{3\times3} & \dots & \dots & \mathbf{0}_{3\times3} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \lfloor(\hat{\mathbf{p}}_{L_a}(k) - \hat{\mathbf{p}}_{L_a}(k') - \frac{\mathbf{g}}{2}({}^{m-1}_0\Delta t)^2)\times\rfloor - \mathbf{\Xi}_a & {}^{m-1}_0\Delta t\mathbf{I}_3 & \mathbf{0}_{3\times3} & \dots & \mathbf{I}_3 & \dots & \mathbf{0}_{3\times3} \\ \hline \lfloor-\frac{\mathbf{g}}{2}({}^m_0\Delta t)^2\times\rfloor + \mathbf{\Lambda}_m & {}^m_0\Delta t\mathbf{I}_3 & \mathbf{0}_{3\times3} & \dots & \mathbf{I}_3 & \dots & \mathbf{0}_{3\times3} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \lfloor-\frac{\mathbf{g}}{2}({}^{k'}_0\Delta t)^2\times\rfloor + \mathbf{\Lambda}_{k'} & {}^{k'}_0\Delta t\mathbf{I}_3 & \mathbf{0}_{3\times3} & \dots & \dots & \dots & \mathbf{I}_3 \end{array}\right]$$

The last equivalence relationship is obtained by the following column operations: first, we multiply the block columns corresponding those landmarks $\mathbf{l}_j$ that are seen *only* by the marginalized robot poses, each by $-\lfloor(\hat{\mathbf{p}}_{L_j}(k)\times\rfloor$, and add to the first block column. Then, we multiply the block columns corresponding to all other landmarks $\mathbf{l}_j$, each by $-\lfloor(\hat{\mathbf{p}}_{L_j}(k')\times\rfloor - \mathbf{\Xi}_j$, and add to the first block column. If we next multiply the first block column of the above matrix by the matrix $\mathbf{F}$

in (141), and define $\Delta\mathbf{p}_{L_a} = \hat{\mathbf{p}}_{L_a}(k) - \hat{\mathbf{p}}_{L_a}(k')$, we obtain:

$$\mathbf{K}(k,k') \sim \left[\begin{array}{cc|ccccc} \frac{{}_0^0\Delta t^2}{2}\begin{bmatrix}\mathbf{G} & \mathbf{0}_{3\times1}\end{bmatrix} & {}_0^0\Delta t\mathbf{I}_3 & \mathbf{I}_3 & \mathbf{0}_{3\times3} & \dots & \dots & \mathbf{0}_{3\times3} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \frac{{}_0^a\Delta t^2}{2}\begin{bmatrix}\mathbf{G} & \mathbf{0}_{3\times1}\end{bmatrix} + \lfloor\Delta\mathbf{p}_{L_a}\times\rfloor\mathbf{F} - \boldsymbol{\Xi}_a\mathbf{F} & {}_0^{m-1}\Delta t\mathbf{I}_3 & \mathbf{0}_{3\times3} & \dots & \mathbf{I}_3 & \dots & \mathbf{0}_{3\times3} \\ \hline \frac{{}_0^a\Delta t^2}{2}\begin{bmatrix}\mathbf{G} & \mathbf{0}_{3\times1}\end{bmatrix} + \boldsymbol{\Lambda}_m\mathbf{F} & {}_0^m\Delta t\mathbf{I}_3 & \mathbf{0}_{3\times3} & \dots & \mathbf{I}_3 & \dots & \mathbf{0}_{3\times3} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \frac{{}_0^{k'}\Delta t^2}{2}\begin{bmatrix}\mathbf{G} & \mathbf{0}_{3\times1}\end{bmatrix} + \boldsymbol{\Lambda}_{k'}\mathbf{F} & {}_0^{k'}\Delta t\mathbf{I}_3 & \mathbf{0}_{3\times3} & \dots & \dots & \dots & \mathbf{I}_3 \end{array}\right]$$

$$\sim \left[\begin{array}{cc|ccccc} {}_0^0\Delta t^2\begin{bmatrix}\mathbf{G} & \mathbf{0}_{3\times1}\end{bmatrix} & {}_0^0\Delta t\mathbf{I}_3 & \mathbf{I}_3 & \mathbf{0}_{3\times3} & \dots & \dots & \mathbf{0}_{3\times3} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ {}_0^a\Delta t^2\begin{bmatrix}\mathbf{G} & \mathbf{0}_{3\times1}\end{bmatrix} + 2\lfloor\Delta\mathbf{p}_{L_a}\times\rfloor\mathbf{F} - 2\boldsymbol{\Xi}_a\mathbf{F} & {}_0^{m-1}\Delta t\mathbf{I}_3 & \mathbf{0}_{3\times3} & \dots & \mathbf{I}_3 & \dots & \mathbf{0}_{3\times3} \\ \hline {}_0^a\Delta t^2\begin{bmatrix}\mathbf{G} & \mathbf{0}_{3\times1}\end{bmatrix} + 2\boldsymbol{\Lambda}_m\mathbf{F} & {}_0^m\Delta t\mathbf{I}_3 & \mathbf{0}_{3\times3} & \dots & \mathbf{I}_3 & \dots & \mathbf{0}_{3\times3} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ {}_0^{k'}\Delta t^2\begin{bmatrix}\mathbf{G} & \mathbf{0}_{3\times1}\end{bmatrix} + 2\boldsymbol{\Lambda}_{k'}\mathbf{F} & {}_0^{k'}\Delta t\mathbf{I}_3 & \mathbf{0}_{3\times3} & \dots & \dots & \dots & \mathbf{I}_3 \end{array}\right] = \mathbf{K}'_{eq}$$

From our analysis in Appendix C.1 we know that all the columns in $\mathbf{K}'_{eq}$ after the third one are linearly independent (they are columns of the matrix $\mathbf{K}_{eq}$ in (142), which is full column rank). Moreover, due to the structure of the terms $\boldsymbol{\Xi}_j$ and $\boldsymbol{\Lambda}_i$ that appear in the first block column, the three column vectors of this block cannot be written as a linear combination of the remaining ones (a detailed proof is tedious but straightforward, and is omitted). Therefore, all the columns in the above matrix are linearly independent, and therefore the rank of $\mathbf{K}(k,k')$ is equal to the number of columns in $\mathbf{K}'_{eq}$, or

$$\text{rank}(\mathbf{K}(k,k')) = 3n+6$$

C.5 Proof of Lemma 9

Relative-position measurements As shown in Appendix C.2, when the robot-to-landmark measurements are direct measurements of the relative position of the landmark with respect to the robot, the matrix $\mathbf{D}$ is a block diagonal matrix with rotation matrices as its diagonal elements, and it is therefore invertible. As a result, $\mathcal{N}(\mathbf{D}(k,k')) = \emptyset$, which shows that

$$\dim(\mathcal{N}(\mathbf{D}(k,k'))\bigcap\mathcal{R}(\mathbf{K}(k,k'))) = 0$$

which is the desired result.

Camera measurements As shown in Appendix C.4, the range of the matrix $\mathbf{K}(k,k')$ is spanned by the columns of the matrix $\mathbf{K}'_{eq}$, which has full column rank. Therefore,

$$\mathcal{N}(\mathbf{D}(k,k'))\bigcap\mathcal{R}(\mathbf{K}(k,k')) = \emptyset \;\Leftrightarrow\; \mathcal{N}(\mathbf{D}(k,k'))\bigcap\mathcal{R}(\mathbf{K}'_{eq}) = \emptyset \;\Leftrightarrow\; \mathbf{D}(k,k')\mathbf{K}'_{eq}\mathbf{a} = \mathbf{0} \text{ only when } \mathbf{a} = \mathbf{0} \quad (177)$$

Let us define the following partitioning of $\mathbf{a}$:

$$\mathbf{a} = \begin{bmatrix} \mathbf{a}_q \\ \mathbf{a}_v \\ \mathbf{a}_1 \\ \vdots \\ \vdots \\ \mathbf{a}_n \end{bmatrix} \quad (178)$$

where each of the $\mathbf{a}_1,\dots,\mathbf{a}_n$, $\mathbf{a}_q$, and $\mathbf{a}_v$ is a 3×1 vector. Following the same steps as in the proof in Appendix C.2, we conclude that the condition $\mathbf{D}(k,k')\mathbf{K}'_{eq}\mathbf{a} = \mathbf{0}$ is equivalent to the conditions:

$$\begin{aligned} {}_0^i\Delta t^2\begin{bmatrix}\mathbf{G} & \mathbf{0}_{3\times1}\end{bmatrix}\mathbf{a}_q + {}_0^i\Delta t\mathbf{a}_v + \mathbf{a}_j &= c_{ij}(\hat{\mathbf{p}}_{L_j}(k) - \hat{\mathbf{p}}_{R_i}(k)), \quad \forall(i,j)\in\mathcal{S}'_m \\ \left({}_0^i\Delta t^2\begin{bmatrix}\mathbf{G} & \mathbf{0}_{3\times1}\end{bmatrix} - 2\lfloor\Delta\mathbf{p}_{L_a}\times\rfloor\mathbf{F} - 2\boldsymbol{\Xi}_a\mathbf{F}\right)\mathbf{a}_q + {}_0^i\Delta t\mathbf{a}_v + \mathbf{a}_j &= c_{ij}(\hat{\mathbf{p}}_{L_j}(k) - \hat{\mathbf{p}}_{R_i}(k)), \quad \forall(i,j)\in\mathcal{S}''_m \end{aligned}$$

$$\left({}^{i}_{0}\Delta t^2\left[\mathbf{G}\quad \mathbf{0}_{3\times 1}\right]+2\mathbf{\Lambda}_i\mathbf{F}\right)\mathbf{a}_q+{}^{i}_{0}\Delta t\mathbf{a}_v+\mathbf{a}_j=c_{ij}(\hat{\mathbf{p}}_{L_j}(k')-\hat{\mathbf{p}}_{R_i}(k')),\quad \forall(i,j)\in\mathcal{S}_a$$

where $\mathcal{S}'_m$ denotes the set of indices (i,j), that correspond to measurements of landmarks observed *only* from marginalized states, while $\mathcal{S}'_m$ denotes the set of indices (i,j), that correspond to measurements of landmarks observed *both* by marginalized *and* non-marginalized states. Clearly, the above conditions are very similar to the conditions (150). Specifically, the first set of conditions (landmarks that are observed only in marginalized states), can be rewritten as:

$${}^{i}_{0}\Delta t^2\mathbf{G}\mathbf{a}_{q(1:2)}+{}^{i}_{0}\Delta t\mathbf{a}_v+\mathbf{a}_j=c_{ij}(\hat{\mathbf{p}}_{L_j}(k)-\hat{\mathbf{p}}_{R_i}(k)),\quad \forall(i,j)\in\mathcal{S}'_m$$

where $\mathbf{a}_{q(1:2)}$ is a vector containing the first two elements of $\mathbf{a}_q$. The above equation is identical to (150), and therefore, using the result of Appendix C.2 we can prove that barring degenerate cases, $\mathbf{a}_{q(1:2)}=\mathbf{0}_{2\times 1}$, $\mathbf{a}_v=\mathbf{0}_{2\times 1}$, and $\mathbf{a}_j=\mathbf{0}$ for all landmarks in this set. Then the remaining conditions become:

$$\begin{aligned}(-2\lfloor\Delta\mathbf{p}_{L_a}\times\rfloor-2\mathbf{\Xi}_a)\,\mathbf{g}a_{q3}+\mathbf{a}_j&=c_{ij}(\hat{\mathbf{p}}_{L_j}(k)-\hat{\mathbf{p}}_{R_i}(k)),\quad \forall(i,j)\in\mathcal{S}''_m\\ 2\mathbf{\Lambda}_i\mathbf{g}a_{q3}+\mathbf{a}_j&=c_{ij}(\hat{\mathbf{p}}_{L_j}(k')-\hat{\mathbf{p}}_{R_i}(k')),\quad \forall(i,j)\in\mathcal{S}_a\end{aligned}$$

where a_{q3} is the third element of the vector $\mathbf{a}_q$, and we used the definition of the matrix $\mathbf{F}$ in (141). If a landmark is seen at least two times, from the second condition above we obtain:

$$\begin{aligned}2\mathbf{\Lambda}_i\mathbf{g}a_{q3}+\mathbf{a}_j&=c_{ij}(\hat{\mathbf{p}}_{L_j}(k')-\hat{\mathbf{p}}_{R_i}(k'))\\ 2\mathbf{\Lambda}_{i'}\mathbf{g}a_{q3}+\mathbf{a}_j&=c_{i'j}(\hat{\mathbf{p}}_{L_j}(k')-\hat{\mathbf{p}}_{R'_i}(k'))\\ \Rightarrow 2(\mathbf{\Lambda}_i-\mathbf{\Lambda}_{i'})\mathbf{g}a_{q3}&=c_{ij}(\hat{\mathbf{p}}_{L_j}(k')-\hat{\mathbf{p}}_{R'_i}(k'))-c_{i'j}(\hat{\mathbf{p}}_{L_j}(k')-\hat{\mathbf{p}}_{R'_i}(k'))\end{aligned}$$

In general, the vectors $(\mathbf{\Lambda}_i-\mathbf{\Lambda}_{i'})\mathbf{g}$, $(\hat{\mathbf{p}}_{L_j}(k')-\hat{\mathbf{p}}_{R'_i}(k'))$, and $(\hat{\mathbf{p}}_{L_j}(k')-\hat{\mathbf{p}}_{R'_i}(k'))$ are linearly independent, and therefore the above condition is only met when $a_{q3}=c_{ij}=c_{i'j}=0$. Therefore, we proved that $\mathbf{a}_q=\mathbf{0}_{3\times 1}$. Using this result, we can now show that all $\mathbf{a}_j$ $j=1\ldots n$ must equal zero. In conclusion, the condition $\mathbf{D}(k,k')\mathbf{K}'_{eq}\mathbf{a}=\mathbf{0}$ is satisfied only when $\mathbf{a}=\mathbf{0}$, which is the desired result.

References

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

[2] S. I. Roumeliotis, A. E. Johnson, and J. F. Montgomery, "Augmenting inertial navigation with image-based motion estimation," in *Proceedings of the IEEE International Conference on Robotics and Automation*, Washington D.C, May 2002, pp. 4326–33.

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

[4] 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.

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

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

[7] A. I. Mourikis and S. I. Roumeliotis, "A multi-state constraint Kalman filter for vision-aided inertial navigation," in *Proceedings of the IEEE International Conference on Robotics and Automation*, Rome, Italy, Apr. 2007, pp. 3565–3572.

[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 *Proceedings of the Photogrammetric Computer Vision Conference*, 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 *Proceedings of the International 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 *Proceedings of the IEEE Computer Society Conference 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 *Proceedings of the IEEE/RSJ International Conference 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. P. Huang, A. I. Mourikis, and S. I. Roumeliotis, "Analysis and improvement of the consistency of extended Kalman filter-based SLAM," in *Proceedings of the IEEE International Conference on Robotics and Automation*, Pasadena, CA, May 2008, pp. 473–479.

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

[19] C. Meyer, *Matrix analysis and applied linear algebra: solutions manual*. Society for Industrial and Applied Mathematics, 2000.

[20] N. Trawny and S. Roumeliotis, "Indirect Kalman filter for 6D pose estimation," *University of Minnesota, Dept. of Comp. Sci. & Eng., Tech. Rep*, vol. 2, 2005.

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