Institute for Aerospace Studies
UNIVERSITY OF TORONTO

Invariant EKF-SLAM on the Starry Night Dataset

AER 1513: State Estimation

Abhinav Grover
<abhinav.grover@robotics.utias.utoronto.ca>

December 31, 2019

Abstract

Simultaneous Localization and Mapping (SLAM) has been a highly research problem and the techniques that solve it have undergone huge improvements in the last two decades. One of the most popular technique is the Extended Kalman filtering (EKF) approach, but it suffers from the problem of inconsistency. Out of the many ways developed to solve this problem, the Invariant Kalman filter based approach is quite interesting and offers provable mathematical guarantees. This project implements an Invariant EKF-SLAM method by representing the robot pose as a member of the special euclidean Lie group. The normalized estimation error square (NEES) metric is also computed which reveals that the given implementation, though works, is over-confident. The inconsistency appears during the process of loop closure and experiments are suggested for further investigation.

1 Introduction

Simultaneous Localization and Mapping, also known as SLAM, is a fundamental problem in the field of state estimation [6]. The SLAM problem exists in multiple domains with different names, for example, the SLAM problem is known as bundle adjustment in domain of image processing and structure from motion in computer vision. There are many methods that have solved this problem out of which the Extended Kalman Filter (EKF) based approach is quite popular due to its low computational complexity. Surprisingly, it is known that the EKF based SLAM approach is inherently inconsistent [9] [8] and there have been many methods that claim to have solved the issue of inconsistency [7] [2]. In [2] it is claimed that the invariant EKF approach solves the inconsistency issues. This project attempts to create a novel Invariant EKF algorithm which is highly based on the concept of [3] and evaluate its accuracy as well as consistency on the real-world 'Starry Night' dataset [5].

It is worth noting that some of the code written for this project overlaps with the code written for the Assignment 3 in the same course. Functions such as Lie group functions have been reused for the project deliverable and are provided with the code here: https://github.com/abhitoronto/InvaraintEKFSLAM.

2 Background

In Invariant kalman filtering methods, the state space is assumed to a be matrix lie group. But, it is the choice of the Lie group structure along with the system model allow the derivation of a kalman filter [4]. Matrix Lie groups, such as the Special Orthogonal group ($\mathrm{SO}(3)$), is capable of representing all valid rotation matrices while another matrix Lie group called the Special Euclidean group ($\mathrm{SE}(3)$) is capable of representing all possible valid transformation matrices [3]. In a SLAM setup where the both the robot pose as well as the landmark location are part of the estimated state, the robot pose can be represented as a matrix $\in \mathrm{SE}(3)$ while the landmark locations can be treated as 3D euclidean vectors.

The 'Starry Night' dataset is a real-world dataset collected for the "AER 1513: State Estimation for Robotics" course, such that the assignment contains the task of performing robot pose estimation using the special euclidean Lie group and a least square optimization approach. This dataset is well-suited for

the objectives of this project as it contains all the elements to perform SLAM, and moreover the $\mathrm{SE}(3)$ Lie group structure along with the system model are compatible to perform an Invariant EKF based SLAM.

2.1 Problem Setup

The experimental setup for the starry night dataset consists of a stereo camera with an Inertial Measurement Unit (IMU) rigidly attached to the camera. The map consists of 20 highly reflective markers in a dark background. Reflective markers have also been attached to the camera, and the Vicon motion capture system is used to capture the ground-truth trajectory of the camera as well as the landmarks on the map. The 3 sensors log their data at varying frequencies, which implies that data synchronization is necessary. Fortunately, the data-points in the given dataset have been time-synchronised, as explained in the assignment document [5]. Furthermore, the raw stereo images have already been processed such that feature detection and matching between the left and the right camera image is done, and the available data is the pixel location of unique features visible in both images.

As a visual representation of the problem, figure 1 shows the 3 reference frames being used in this setup. Here the initial frame ($\underrightarrow{\mathcal{F}}_i$) is the global coordinate frame, the vehicle frame ($\underrightarrow{\mathcal{F}}_v$) is the frame of reference for the robot and is attached to the IMU, and ($\underrightarrow{\mathcal{F}}_c$) is the reference frame located at the left camera which is one of the two axis aligned stereo cameras. The translation $\underrightarrow{\rho}^{vc}$ and the rotation $\mathbf{C}_{cv}$ between the vehicle and the camera frame have been pre-computed during calibration and are also given.

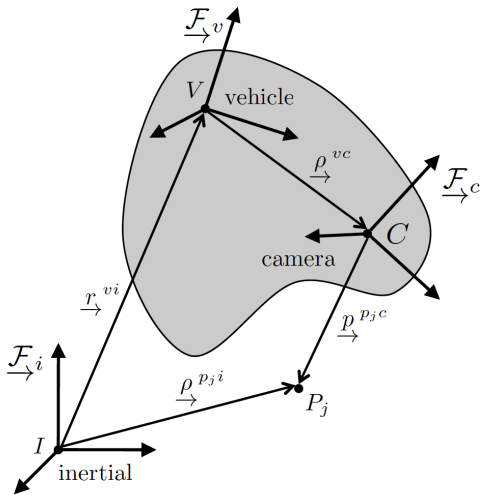

Figure 1: Frame of references used in the problem setup [5]

3 Methodology

3.1 Motion and Measurement Models

In a SLAM setup the state of the filter includes the robot pose as well as the landmark locations. Thus, the state at a given time-step k is represented as $\mathbf{X}_k = \left\{\mathbf{T}_k, \mathbf{f}_k^1, \ldots, \mathbf{f}_k^M\right\}$ where $\mathbf{T}_k \in \mathrm{SE}(3)$ represents the robot pose and $\mathbf{f}_k^j \in \mathbb{R}^3$ is the coordinates of the j^{th} landmark in $\underrightarrow{\mathcal{F}}_i$. The motion of the robot is captured in the IMU data which includes translational velocity $\mathrm{v}_{v_k}^{v_k i}$ and rotational velocity $\boldsymbol{\omega}_{v_k}^{v_k i}$, which when multiplied with the δt_k is the robot odometry. These odometry values at time-step k update the robot pose, while the landmark position stays constant. This comprises the motion model for this filtering setup and has been shown in Equation 1 and 2.

$$\mathbf{T}_k = \boldsymbol{\Xi}_k \mathbf{T}_{k-1} \tag{1}$$

$$\mathbf{f}_k^j = \mathbf{f}_{k-1}^j + \delta \mathbf{d}_k^j \tag{2}$$

where,

$$\boldsymbol{\Xi}_k = \mathbf{exp}\left(\left(\delta t_k \begin{bmatrix} \mathbf{v}_{v_k}^{v_k i} \\ \boldsymbol{\omega}_{v_k}^{v_k i} \end{bmatrix} + \delta \mathbf{w}_k\right)^{\wedge}\right) \qquad \delta t_k = t_k - t_{k-1}$$

$$\delta \mathbf{w}_k \backsim \mathcal{N}(0, \mathbf{Q}_k^w) \qquad \delta \mathbf{d}_k^j \backsim \mathcal{N}(0, \mathbf{Q}_k^{d^j})$$

$$\mathbf{exp}(^{\wedge}) : \mathfrak{se}(3) \rightarrow \mathrm{SE}(3)$$

Here $\mathbf{Q}_k^w$ and $\mathbf{Q}_k^{d^j}$ are the covariances of the combined robot odometry and the landmark positions, respectively.

As mentioned in the previous section, the measurement values $\mathbf{y}_k^j$ are pixel locations of the same landmark in both left and right stereo camera. If the current pose of the robot $\mathbf{T}_k$, current location of the landmarks $\mathbf{f}_k^j$ are known in the inertial frame, and the intrinsic parameters of the camera are known, the pixel-values for the visible landmarks can be calculated using the motion model given in Equation 3.

$$\mathbf{y}_k^j := \begin{bmatrix} u_l \\ v_l \\ u_r \\ v_r \end{bmatrix} = \mathbf{g}(\mathbf{z}(\mathbf{T}_k, \mathbf{f}_k^j)) + \delta \mathbf{n}_k^j \tag{3}$$

given that,

$$\mathbf{z}(\mathbf{T}_k, \mathbf{f}_k^j) = \mathbf{D}^T \mathbf{T}_{cv} \mathbf{T}_k \mathbf{p}_k^j \qquad \mathbf{g}(\begin{bmatrix} x \\ y \\ z \end{bmatrix}) = \begin{bmatrix} \frac{f_u x}{z} + c_u \\ \frac{f_v y}{z} + c_v \\ \frac{f_u (x-b)}{z} + c_u \\ \frac{f_v y}{z} + c_v \end{bmatrix}$$

where,

$$\mathbf{D} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \qquad \mathbf{p}_k^j = \begin{bmatrix} \mathbf{f}_k^j \\ 1 \end{bmatrix} \qquad \delta\mathbf{n}_k^j \backsim \mathcal{N}(0, \mathbf{R}_k^j) \qquad \mathbf{T}_{cv} = \begin{bmatrix} \mathbf{C}_{cv} & -\mathbf{C}_{cv}\boldsymbol{\rho}_v^{cv} \\ \mathbf{0}^T & 1 \end{bmatrix}$$

Here, the symbols f_u, f_v, c_u, c_v and b are intrinsic camera parameters and are provided with the dataset [5]. $\mathbf{R}_k^j$ is the covariance of recorded camera pixels and $\mathbf{T}_{cv}$ is the transformation matrix from the vehicle frame to the camera frame of reference.

3.2 Covariance Values

The dataset document [5] provides histograms of the velocity measurement error of the IMU as well as the stereo camera pixel error. These values have been calculated using the ground-truth data available through the Vicon motion capture system. Based on the profile of the error histograms in the document, the sensor noise is assumed to be zero mean Gaussian, as shown in equation 1, 2 and 3. Since equation 1 requires variance measurements for vehicle odometry, the given vehicle velocity variances must be multiplied with δt_k^2 at every time-step to extract the required variance values. Moreover, a low variance of 10^{-6} m^2 has been chosen for the landmark coordinates, since the landmarks coordinates don't change during vehicle motion.

3.3 Invariant EKF - SLAM

The above defined motion and measurement models are necessary for the EKF-SLAM algorithm. The corresponding SLAM algorithm is shown below. Here the goal is to calculate an estimate of the robot pose and landmark location along with their respective covariances for the current time step. $\mathbf{P}_k$ represents the

combined covariance of the complete state $\mathbf{X}_k$ being estimated.

$$\text{predictor}: \begin{cases} \check{\mathbf{T}}_k &= \boldsymbol{\Xi}_k \hat{\mathbf{T}}_{k-1} \quad \text{with} \quad \delta\mathbf{w}_k = \mathbf{0} \\ \check{\mathbf{f}}_k &= \hat{\mathbf{f}}_{k-1} \\ \check{\mathbf{P}}_k &= \mathbf{F}_{k-1}\hat{\mathbf{P}}_{k-1}\mathbf{F}_{k-1}^T + \mathbf{Q}_k \end{cases} \tag{4}$$

$$\text{innovation}: \begin{cases} \mathbf{K}_k &= \check{\mathbf{P}}_k \mathbf{G}_k^T (\mathbf{G}_k \check{\mathbf{P}}_k \mathbf{G}_k^T + \mathbf{R}_k)^{-1} \\ \mathbf{I}_k &:= \begin{bmatrix} \mathbf{I}_{\mathbf{T},k}^T & \mathbf{I}_{\mathbf{f},k}^T \end{bmatrix}^T = \mathbf{K}_k(\mathbf{y}_k - \check{\mathbf{y}}_k) \end{cases} \tag{5}$$

$$\text{corrector}: \begin{cases} \hat{\mathbf{T}}_k &= \mathbf{exp}(\mathbf{I}_{\mathbf{T},k}{}^{\wedge})\check{\mathbf{T}}_k \\ \hat{\mathbf{f}}_k &= \check{\mathbf{f}}_k + \mathbf{I}_{\mathbf{f},k} \\ \hat{\mathbf{P}}_k &= (\mathbf{1} - \mathbf{K}_k\mathbf{G}_k)\check{\mathbf{P}}_k \end{cases} \tag{6}$$

where,

$$\mathbf{f}_k = \begin{bmatrix} \mathbf{f}_k^1 \\ \vdots \\ \mathbf{f}_k^M \end{bmatrix} \qquad \mathbf{Q}_k = \begin{bmatrix} \mathbf{Q}_k^w & & & \\ & \mathbf{Q}_k^{d^1} & & \\ & & \ddots & \\ & & & \mathbf{Q}_k^{d^M} \end{bmatrix} \qquad \mathbf{R}_k = \begin{bmatrix} \mathbf{R}_k^1 & & \\ & \ddots & \\ & & \mathbf{R}_k^M \end{bmatrix}$$

$$\mathbf{y}_k = \begin{bmatrix} \mathbf{y}_k^1 \\ \vdots \\ \mathbf{y}_k^M \end{bmatrix} \qquad \check{\mathbf{y}}_k = \begin{bmatrix} \check{\mathbf{y}}_k^1 \\ \vdots \\ \check{\mathbf{y}}_k^M \end{bmatrix} = \begin{bmatrix} \mathbf{g}(\mathbf{z}(\check{\mathbf{T}}_k, \check{\mathbf{f}}_k^1)) \\ \vdots \\ \mathbf{g}(\mathbf{z}(\check{\mathbf{T}}_k, \check{\mathbf{f}}_k^M)) \end{bmatrix}$$

The matrices $\mathbf{F}_k$ and $\mathbf{G}_k$ are the jacobians of the measurement and motion models, respectively, with respect to the input state at time step k. The perturbation in the robot pose $\mathbf{T}_k \in \mathrm{SE}(3)$ is defined as $\mathbf{T'}_k = \mathbf{exp}(\delta\boldsymbol{\xi}^{\wedge})\mathbf{T}_k$ where $\delta\boldsymbol{\xi} \in \mathfrak{se}(3)$ is the perturbation. According to the equation 8.127 in the state estimation book [3], the jacobian for the robot pose motion model (Equation 1) is the Adjoint [3] of the odometry transformation. Whereas, the jacobian matrix of the landmark motion model is clearly the identity matrix. The resultant combined jacobian $\mathbf{F}_k$ is given in equation 7. Similarly, the measurement

jacobian $\mathbf{G}_k$ is given in the equation 8.

$$\mathbf{F}_{k-1} = \begin{bmatrix} \mathbf{F}_{k-1}^{\mathbf{T}} & & & \\ & \mathbf{F}_{k-1}^{\mathbf{f}^1} & & \\ & & \ddots & \\ & & & \mathbf{F}_{k-1}^{\mathbf{f}^M} \end{bmatrix} = \begin{bmatrix} Ad(\mathbf{\Xi}_k) & & & \\ & \mathbf{1} & & \\ & & \ddots & \\ & & & \mathbf{1} \end{bmatrix} \tag{7}$$

$$\mathbf{G}_k = \begin{bmatrix} \mathbf{G}_{k,\mathbf{T}}^1 & \mathbf{G}_{k,\mathbf{f}}^1 & & \\ \vdots & & \ddots & \\ \mathbf{G}_{k,\mathbf{T}}^M & & & \mathbf{G}_{k,\mathbf{f}}^M \end{bmatrix} \tag{8}$$

where,

$$\mathbf{G}_{k,\mathbf{T}}^j = \mathbf{g}(\mathbf{z}(\check{\mathbf{T}}_k, \check{\mathbf{f}}_k^j))\mathbf{D}^T\mathbf{T}_{cv}(\check{\mathbf{T}}_k\check{\mathbf{p}}_j)^{\odot} \qquad \mathbf{G}_{k,\mathbf{f}}^j = \mathbf{g}(\mathbf{z}(\check{\mathbf{T}}_k, \check{\mathbf{f}}_k^j))\mathbf{D}^T\mathbf{T}_{cv}\check{\mathbf{T}}_k\mathbf{D}$$

3.4 Error Calculation

In order to evaluate the consistency of the estimation algorithm, a widely used metric is the Normalized Estimation Error Squared (NEES) [1]. NEES value scales the error according to the estimation covariance (Equation 9) and provided an appropriate picture of the estimation consistency. An estimator with the NEES value greater than 1 is considered optimistic, whereas for a value less than 1 the estimator is considered pessimistic.

$$\epsilon_{nees} = \delta\mathbf{x}_{error}^T\mathbf{P}_x^{-1}\delta\mathbf{x}_{error} \tag{9}$$

The method to calculate the translational and rotational error for the robot are given in the dataset document [5] while the positional error of the landmarks is a simple vector difference.

4 Results and Discussion

This section discusses the results of performing EKF-SLAM on the Starry Night dataset. Figure 2 shows the distribution of the visible landamarks across the time horizon of the dataset.

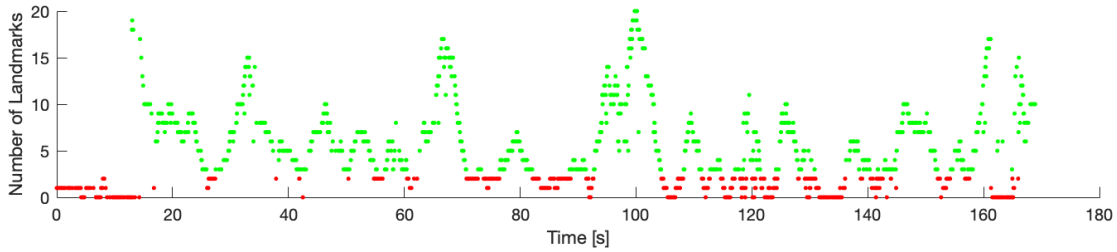

Figure 2: Number of Landmarks at given timesteps. Green dots imply that 3 or more landmarks are visible. The dots are red otherwise.

In order to maximize the visibility of landmarks, the EKF-SLAM algorithm is run from 0 to 50 seconds of the dataset with the landmark locations randomly initialized about the ground-truth with a standard deviation of 20 mm. Figure 3 shows the ground-truth trajectory for the mentioned period of time, as well as, the output robot trajectory along with the estimated landmark locations in red. It is clear that the plot of the estimated location is much more irregular, while the ground-truth plot is a lot more smooth.

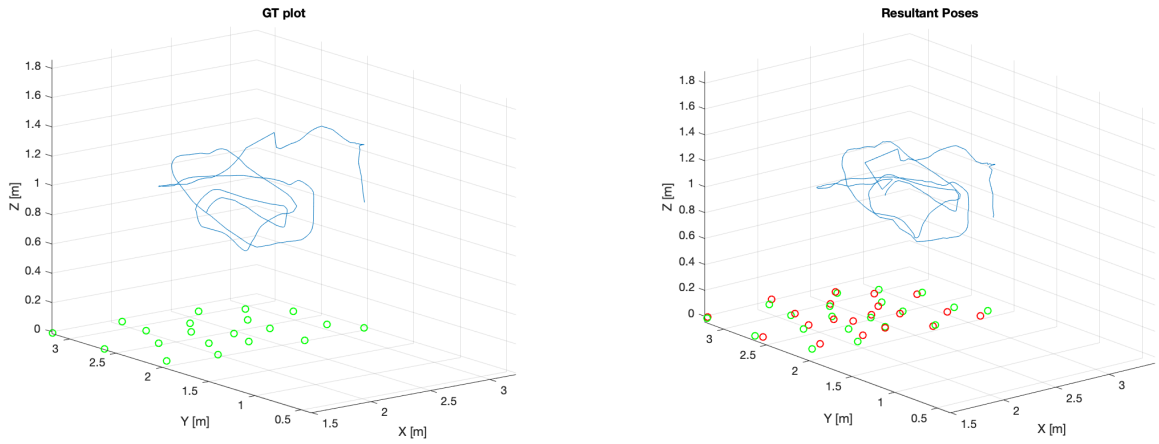

Figure 3: Ground-truth robot trajectory (left) and the resultant trajectory (Right) after applying the EKF-SLAM algorithm for the 'Starry Night' dataset

The translational and rotational accuracy of the estimated robot poses is shown in figure 5. The x coordinate of the robot position consistently increases in error while its variance remains low. This hints towards estimator inconsistency. On the other hand, rotational error appear to lie within the 3σ confidence bounds. Another interesting observation is that, around the 12 seconds mark, there is a drastic drop in the estimator variance, which indicates a loop closure like situation due to a sudden increase in the number of visible landmarks (figure 2).

In order to evaluate the accuracy of the landmark estimation, the root mean square (RMS) error of each landmark coordinate has been plotted against time in figure 4. This plot basiclaly represents the average error of the estimated landmarks. It appears that after the above mentioned loop closure, the estimator becomes over-confident about the x and y coordinates of the landmarks while the error values increase and reach a limit.

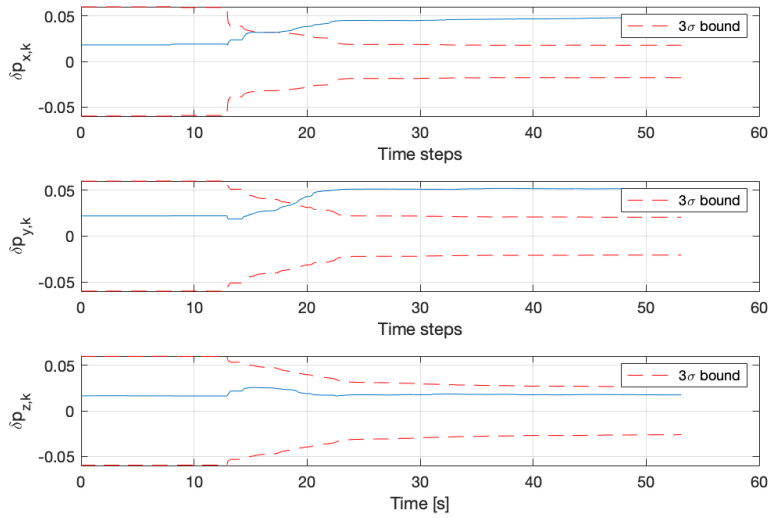

Figure 4: RMS positional error plots for 20 landmarks with the 3σ bounds created with the average variance of landmark coordinates.

As mentioned in earlier sections, the NEES is a widely accepted measure to evaluate the consistency of an estimator. Figure 6 shows the NEES metric plots for the robot position and orientation along with the landmark positions. The error values in all the plots exceed the NEES value of 1 which implies that the estimator is highly over-confident. The error plots in figure 5 and 4 also hinted to-

wards this result. It is worth nothing that all the code to generate these plots is available online at https://github.com/abhitoronto/InvaraintEKFSLAM.

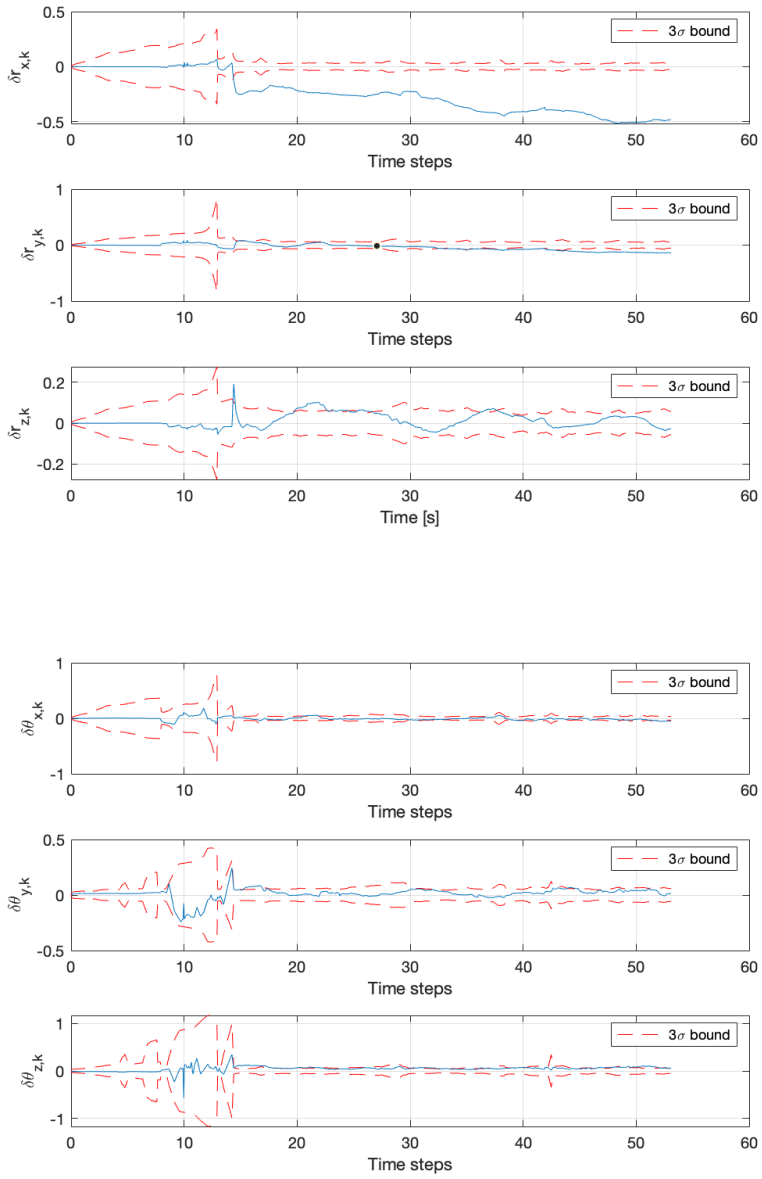

Figure 5: Translational (Top) and rotational (Bottom) error plots

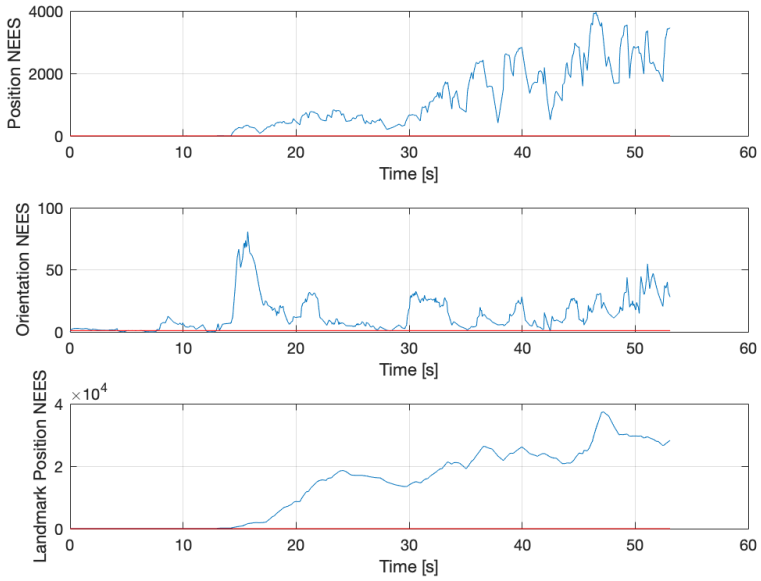

Figure 6: NEES (blue) plot vs time for the robot position (top), robot orientation (middle) and the landmark positions (bottom). The line is red is a horizontal like passing through 1.

5 Conclusion and Future Work

This report outlines the methodology to implement an Invariat EKF-SLAM approach on the 'Starry Night' dataset and attempts to evaluate the consistency of the technique. The technique results in an inconsistent estimator which is highly over-confident in the robot position as well as the landmark positions. Further investigation of the consistency properties of the technique is required to identify the root-cause. Suggested experiments would be to run the estimator on a dataset with a clear loop closure along with less jerky odometry. Another approach would be to follow the consistency tests used in [2] for the RI-EKF SLAM.

References

[1] Practical measures and test for credibility of an estimator. *Proc. Workshop on Estimation, Tracking, and . . .*, pages 481—-495, 2001.

[2] Convergence and Consistency Analysis for a 3-D Invariant-EKF SLAM. *IEEE Robotics and Automation Letters*, 2(2):733–740, 2017.

[3] Timothy D Barfoot. *State Estimation for Robotics*. Cambridge University Press, 2017.

[4] Axel Barrau and Silvère Bonnabel. Invariant Kalman Filtering. *Annual Review of Control, Robotics, and Autonomous Systems*, 1(1):237–257, may 2018.

[5] Timothy David Barfoot. State Estimation for Aerospace Vehicles-AER1513 Course Assignments-A LEAST-SQUARES TRILOGY UTIAS. Technical report, 2011.

[6] M. W.M. Gamini Dissanayake, Paul Newman, Steven Clark, Hugh F. Durrant-Whyte, and M. Csorba. A solution to the simultaneous localization and map building (SLAM) problem. *IEEE Transactions on Robotics and Automation*, 17(3):229–241, jun 2001.

[7] Guoquan P. Huang, Anastasios I. Mourikis, and Stergios I. Roumeliotis. Observability-based rules for designing consistent EKF SLAM estimators. In *International Journal of Robotics Research*, volume 29, pages 502–528, apr 2010.

[8] Shoudong Huang and Gamini Dissanayake. Convergence and consistency analysis for extended Kalman filter based SLAM. *IEEE Transactions on Robotics*, 23(5):1036–1049, oct 2007.

[9] S. J. Julier and J. K. Uhlmann. A counter example to the theory of simultaneous localization and map building. In *Proceedings - IEEE International Conference on Robotics and Automation*, volume 4, pages 4238–4243, 2001.