PRECISE VEHICLE LOCALIZATION USING FUSION OF MULTIPLE SENSORS FOR SELF-DRIVING

Undergraduate graduation project report submitted in partial fulfillment of
the requirements for the
Degree of Bachelor of Science of Engineering
in

The Department of Electronic & Telecommunication Engineering
University of Moratuwa.

Supervisor:
Dr. Peshala Jayasekara

Group Members:
S. P. Dissanayake (160134U)
G. M. C. P. Jayathissa (160188L)
S. H. P. H. P. Jayawardana (160246N)
A. S. R. Kumara (160318M)

March, 2021

Approval of the Department of Electronic & Telecommunication Engineering

. . . .
Head, Department of Electronic & Telecommunication Engineering

This is to certify that I/we have read this project and that in my/our opinion it is fully adequate, in scope and quality, as an Undergraduate Graduation Project.

Supervisor: Dr. Peshala Jayasekara

Signature: . . . .

Date: . . . .

DECLARATION

This declaration is made on March 21, 2021.

Declaration by Project Group

We declare that the dissertation entitled ”Precise Vehicle Localization Using Fusion of Multiple Sensors for Self-Driving” and the work presented in it are our own. We confirm that:

- this work was done wholly or mainly in candidature for a B.Sc. Engineering degree at this university,
- where any part of this dissertation has previously been submitted for a degree or any other qualification at this university or any other institute, has been clearly stated,
- where we have consulted the published work of others, is always clearly attributed,
- where we have quoted from the work of others, the source is always given,
- with the exception of such quotations, this dissertation is entirely our own work,
- we have acknowledged all main sources of help,

. . . .
Date

. . . .
S. P. Dissanayake (160134U)

. . . .
G. M. C. P. Jayathissa (160188L)

. . . .
S. H. P. H. P. Jayawardana (160246N)

. . . .
A. S. R. Kumara (160318M)

DECLARATION BY SUPERVISOR

I have supervised and accepted this dissertation for the submission of the degree.

. . . .

Dr. Peshala Jayasekara | Date

ABSTRACT

PRECISE VEHICLE LOCALIZATION USING FUSION OF MULTIPLE SENSORS FOR SELF-DRIVING

Group Members: S. P. Dissanayake, G. M. C. P. Jayathissa, S. H. P. H. P. Jayawardana, A. S. R. Kumara

Supervisor: Dr. Peshala Jayasekara

Keywords: Self-Driving, State Estimation, Localization, Sensor Fusion, Bayesian Filters.

This project focuses on creating a mechanism for estimating the state of a self-driving vehicle, including its location, speed and orientation, relative to a coordinate frame fixed to earth. We expect to achieve this using data from sensors such as Inertial Measurement Unit (IMU), Global Navigation Satellite System (GNSS) receivers, stereo camera pairs and Light Detection and Ranging (LiDAR) sensors. The main objective is to deliver a well-documented software stack which includes the state estimator running on Robot Operating System (ROS). The estimator should be capable of providing uninterrupted state estimations with enough accuracy and frequency to facilitate self-driving.

The main drawback observed in current state-of-the-art work is, the dependency of the solution on pre-generated highly-detailed maps of different forms, which in-turn reduces the scalability of the solution. This dependency reduces the feasibility of those solutions in the long run due to the fact that it is hard to maintain such highly-detailed maps in midst of constantly and unexpectedly changing environments, prevailing in countries such as Sri Lanka. It is the intention of this project to mitigate this dependency through means of improving the state estimation algorithm. We also intend to implement the solution in a modularized architecture to facilitate easy modifications, which in-turn will allow the solution to be used in different applications.

While self-driving is itself a novel concept in Sri Lankan context, this project aims to facilitate the state estimation under constrained resource availability (such as excluding highly-detailed maps, enhanced GNSS technologies such as Differential Global Positioning System (DGPS) or Real Time Kinematics (RTK) Global Positioning System (GPS), reliable road features such as consistent lane markings and curbs etc.), which is the condition experienced in countries like Sri Lanka.

Other than the self-driving research communities, we expect the outcome of this project will benefit different parties such as robot developers and navigational solution providers, who have similar requirements.

DEDICATION

TODO.

ACKNOWLEDGEMENTS

TODO.

Contents

List of Figures

List of Tables

Chapter 1

INTRODUCTION

1.1 Problem definition and scope

The amount of autonomy in vehicles is divided into six levels and vehicles beyond level three are considered as self-driving. Such vehicles are expected to travel from a starting point to a given destination, with minimal human-driver involvement. Therefore, they need to know their location with a very high accuracy, relative to their immediate environment as well as in a global level. Other than the location itself, it is important to provide details of other state variables of the vehicle such as the speed and the orientation, which will be used by top-level controlling algorithms. This information should be updated uninterruptedly and frequently to preserve accuracy under higher speeds, which is the job of a localization module. The most widely used mechanism for fulfilling this requirement is, fusing data obtained from different sensors such as Global Navigation Satellite System (GNSS), Light Detection and Ranging (LiDAR), Radio Detection and Ranging (RADAR) and cameras to obtain the most probable state using a Bayesian filter. This is known as sensor fusion.

As we have noticed, a main limitation of the existing state-of-the-art work in this regard is the dependency of these solutions on different kinds of existing, accurate feature maps. These maps are used as inputs to the localization module, which reduces the scalability of the solution due to the fact that creating, updating and storing such highly-detailed maps of an entire region or a country is not so feasible. We also note the absence of a detailed work-flow describing complete implementation of a localization module, which in-turn wastes the time and effort of the research community, by having to start from the beginning, all the time.

Therefore, it is the aim of this project to implement a localization module which addresses problems mentioned above, while providing enough accuracy and update frequency to allow self-driving. Mentioning specifically, a sub-meter level positional accuracy is targeted along with a frequency of 70 Hz or more, which will be sufficient for speeds below 50 km h^{-1}[1]. As this work is a part of the top-level project aiming the construction of a fully autonomous vehicle, the system is implemented on Robot Operating System (ROS) to facilitate easy integration with other modules. The solution will initially be tuned and tested using freely available datasets and, eventually, it will be tested using actual sensors. The accuracy will only be evaluated using datasets, by comparing with the provided ground truth data.

While self-driving is itself a novel concept in the Sri Lankan context, this project aims in resolving the dependency of state-of-the-art work on feature maps, thereby allowing accurate localization in unstructured, constantly changing environments. We intend to achieve this goal mainly through improving the data fusion algorithm. Other than the field of self-driving itself, we expect that this mechanism will be useful in different applications such as robotics navigation and navigational equipment development. The project is carried out with the collaboration of Creative Software Private Limited.

1.2 Related work

Localization using multi-sensor fusion is not a brand-new idea. A lot of researches have been done in this area for the past decade. The self-driving car concept was firstly addressed with Military European Land Robot Trial (M-ELROB) and Defense Advance Research Project Agency (DARPA) Urban Challenge competitions in 2006 and 2007[2]. The SmartTer [3] and Stanford Junior [4] are two self-driving car projects who won these competitions. Google's car [5], VisLab's Car [6], Apollo [7] and Autoware Auto [8] are some examples of successful research projects. However, self-driving cars are still in the introductory phase of the product life cycle.

Normally, GNSS, Inertial Measurement Unit (IMU), LiDAR, RADAR and wheel encoders are used to localize a robot. The combination of sensors depends on the design. Different sensors have different shortcomings. GNSS signal may not be available on a covered area, underground, or in a tunnel. Normally GNSS accuracy is about 10m due to satellite orbit and clock errors [9]. In addition, when the vehicle drives next to large structures, GNSS measurement can go wrong due to reflections [3]. GISA, which is a Brazilian platform for autonomous car trials uses Differential Global Positioning System (DGPS) as one of the sensors [2]. It gives better accuracy than normal GNSS. DGPS has also been used by the SmartTer. But standard GNSS is available more often when compared to DGPS because it does not rely on the visibility of geostationary satellites which provide the DGPS corrections [3]. The enhanced GNSS technique known as Real Time Kinematics (RTK) has been used by Wan et al. in their self-driving car for localization [9]. However, this method is also prone to significant errors caused by multipath effects and signal blockages due to its dependency on precision carrier-phase positioning techniques. It is noted that most of the projects use normal GNSS[4] [10][11][12]. LiDAR works well when the environment is full of 3D or texture features, but it fails in open spaces [9]. 3D LiDAR sensors were used by GIZA[2], Wan et al. [9] and Levinson et al. [13], while 2D LiDAR sensors were used by Soloviev [10], [14] and Baldwin and Newman [15]. Erik Ward and John Folkesson used

RADAR as the measurement model input as mentioned in [16]. Both LiDAR and RADAR sensors have the same kind of behavior when they act as measurement model inputs. IMU sensors have been used in almost every project. However, it suffers from the accumulation of integration errors [9]. Wheel encoders have been used along with IMU in [3], but it provides incorrect measurements when wheels slip. As mentioned above, each sensor has its own drawbacks and advantages. Other than these sensors Wan et al. [9], Levinson and Thrun [11], Stanford Junior [4], and Apollo [7] projects have used a pre-built map. Currently, some measurement companies have already begun to prepare map databases for self-driving vehicles [17].

In the works of Wan et al., they have used LiDAR intensity and altitude cues with 3D geometry for their LiDAR based localization module. They have obtained a grid-cell representation of the environment using a single Gaussian distribution to model the environment which involved both the intensity and the altitude. Finally, an Error-State Kalman filter was applied to fuse the data from the sensors. They have achieved 0.05-0.1 m Root Mean Square (RMS) accuracy in both longitudinal and lateral directions [9]. The Stanford Junior which won the second place of DARPA Urban Challenge competition was given a digital map of the road network in the form of a Route Network Definition File (RNDF). The RNDF contained geometric information about lane markings, stop signs, parking lots and special checkpoints. They were specified in GNSS coordinates. Local alignment between the RNDF and the vehicle's current position was estimated using GNSS along with two laser sensor measurements [4]. After the competition, they have upgraded the ground map using GNSS, IMU and Velodyne LiDAR data. Here, every cell was represented as its own Gaussian distribution. By this method, they have achieved a lateral RMS accuracy better than 0.1 m. Levinson and Thrun have localized the vehicle using a probabilistic map. They have modeled the environment as a probabilistic grid whereby every cell is represented as its own Gaussian distribution over remittance values. Furthermore, offline Simultaneous Localization and Mapping (SLAM) was used to align multiple passes of the same environment. Once a map had been built, they have used it to localize the vehicle in real time by representing the likelihood distribution of possible x and y offsets with a 2-dimensional histogram filter. The resulting error after localization has been extremely low, with an RMS value of 0.09 m [11]. A hybrid model with both Kalman filter and particle filter has been proposed by Won et al. in [18] as the sensor fusion algorithm. They have used the particle filter to estimate the orientation and the Kalman filter for estimating the position and velocity. As per the above discussion, it is clear that the most widely used and successful method for localization for self-driving is fusing data obtained from different sensors. These sensors should be selected carefully, so that they have complementary properties and functioning capabilities under different environmental conditions. As an example, a GNSS receiver may function well in an open environment, in which a LiDAR is of very little use. Conversely, a LiDAR can give a very detailed output while in an urban environment, in which a GNSS receiver may fail due to multipath effects and signal blockage. We also note that in almost all the works, a Bayesian filter (Extended Kalman, Particle filter etc.) or a combination of many, has been used as the data fusion algorithm.

Another important deduction that can be made from the above comparison is that most of the state-of-the-art work depends on the assumption of an existing accurate map, which is given as an input to the localization mechanism. Even though this dependency is satisfiable under constrained environments such as competitions, it reduces the scalability of the solution drastically when it comes to unstructured environments, which a self-driving vehicle will be experiencing most frequently under normal operation. This is also the case for countries like Sri Lanka where highly detailed mapping of the entire country is a tedious task due to resource constrains and frequent and unplanned changings of the features. Hence, we find it is extremely important to focus on reaping the maximum localization accuracy possible, from a given combination of sensors.

1.3 Method of investigation and results

After a comprehensive study of literature pertaining to current developments, we decided to use a Bayesian framework of sensor fusion, along with the sensor data from (not limited to) GNSS receiver, IMU, LiDAR, stereo camera pair, magnetometer and wheel odometer. In-order to decide upon the type of the Bayesian filter to be used, several such filters with a minimal complexity were implemented, and accuracies were compared using data from existing datasets. As developing visual odometry (from stereo pair of cameras) and LiDAR odometry algorithms were out of the scope of the project, existing state-of-the-art algorithms were used with minor modifications. A filter framework with stochastic cloning and backward smoothing functionality was implemented in Python. Special attention was given to implement this framework in an easily modifiable manner (mainly, the motion model and state variables), so that, it facilitates experimenting with different filter structures. Output of the framework is compared with the ground truth provided in the datasets to obtain accuracy measures. As per the current state of the project, two main problems have been identified; mechanism to estimate an error covariance matrix for visual/LiDAR odometry algorithms and treating the correlated noise of the GNSS measurements. A detailed discussion of the method of investigation and the results obtained will be carried out in the subsequent chapters.

Chapter 2

METHODOLOGY

2.1 System architecture

As depicted in figure 2.1, we use the pose estimates calculated using stereo images and LiDAR point clouds as relative measurements. ORB SLAM 2 and LegoLOAM algorithms, respectively, are used for this purpose. Furthermore, GNSS measurements and orientation estimated from magnetometer measurements act as positional and rotational absolute measurements to the fusion mechanism. The fusion mechanism is an Error State Extended Kalman Filter (ES-EKF) with 16 variables in the state space and 15 variables in the error state space. The output of the system consists of a state vector including position, orientation and velocity of the vehicle relative to a global frame of reference. Estimated covariance of the error is also provided, from which, the 99% confidence interval (3σ bound) can be derived. Output is compared with the ground truth provided with the dataset being used, to calculate the resultant error margins.

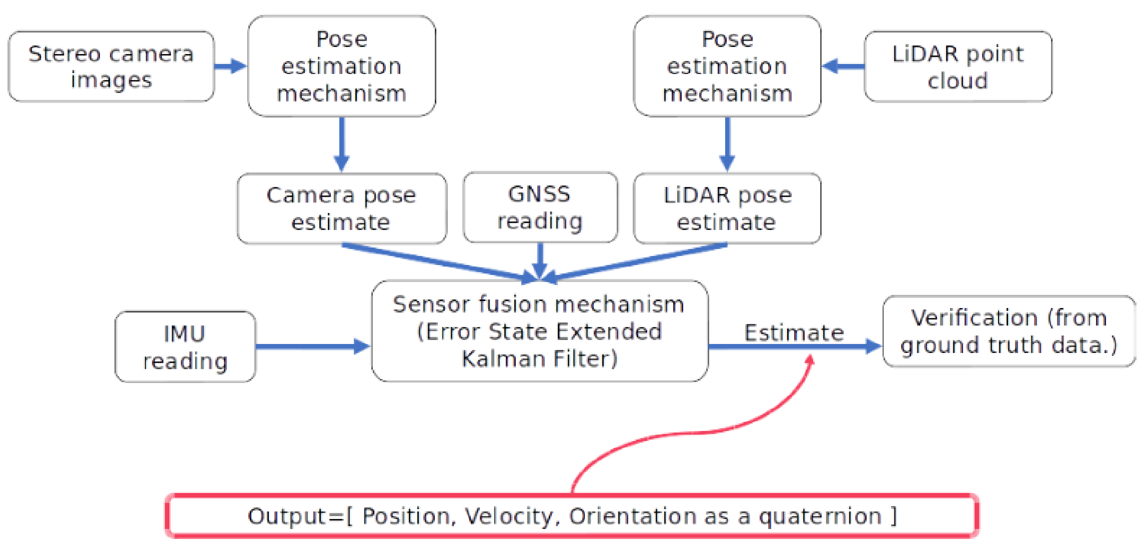

Figure 2.1: System block diagram

The system is implemented on Robot Operating System (ROS), along with evaluation and visualization mechanisms for demonstrating functionality. Main programming language used is Python. Each of the components mentioned in the above discussion will be explained in detail, in the subsequent sections.

2.2 Coordinate frames

Apart from each sensor's own coordinate frame, in which they provide measurements, we define the following coordinate frames which will be used in the rest of this report.

Inertial frame	An earth fixed right-handed rectilinear coordinate frame with x, y and z axes pointing towards East, North and Up directions respectively(ENU frame). The origin of the frame is determined by the information given in the dataset being used.
Body frame	A right-handed rectilinear coordinate frame fixed to the vehicle with x, y and z directions pointing lateral, front and upward directions of the vehicle respectively.

Figure 2.2 illustrates the above-mentioned coordinate frames.

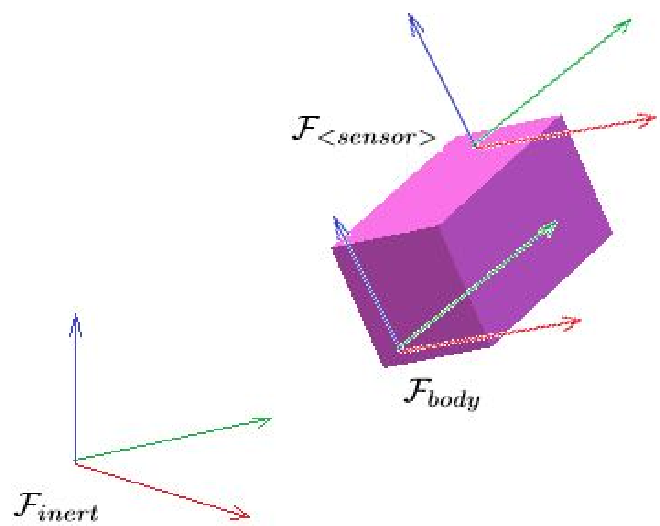

Figure 2.2: Illustration of the coordinate frames. x, y and z axes of each coordinate frame is depicted in red, green and blue colours, respectively. The cuboid represents the vehicle. The sensor is assumed to be mounted on the front side of the roof of the vehicle.

2.3 Sensor fusion mechanism

2.3.1 The Error State Extended Kalman Filter

The ES-EKF acts as the component responsible for fusing sensor data. As mentioned in section 2.1, our ES-EKF currently has 16 nominal state space variables and 15 error state space variables, grouped into 5 sub-vectors, as listed below;

$$\text{Nominal state vector: } \mathbf{x} = \begin{bmatrix} \mathbf{p} \\ \mathbf{v} \\ \mathbf{q} \\ \boldsymbol{a_b} \\ \boldsymbol{\omega_b} \end{bmatrix} \tag{2.1}$$

where

$$\begin{aligned} \mathbf{p} &= (p_x, p_y, p_z) \text{ position relative to the inertial frame} \\ \mathbf{v} &= (v_x, v_y, v_z) \text{ velocity relative to the inertial frame} \\ \mathbf{q} &= (q_w, q_x, q_y, q_z) \text{ quaternion relative to the inertial frame} \\ \boldsymbol{a_b} &= (a_{bx}, a_{by}, a_{bz}) \text{ acceleration biases of the IMU relative to the body frame} \\ \boldsymbol{\omega_b} &= (\omega_{bx}, \omega_{by}, \omega_{bz}) \text{ angular velocity biases of the IMU relative to the body frame} \end{aligned} \tag{2.2}$$

and

$$\text{Error state vector: } \boldsymbol{\delta}\mathbf{x} = \begin{bmatrix} \boldsymbol{\delta}\mathbf{p} \\ \boldsymbol{\delta}\mathbf{v} \\ \boldsymbol{\delta\theta} \\ \boldsymbol{\delta a_b} \\ \boldsymbol{\delta\omega_b} \end{bmatrix}. \tag{2.3}$$

Here, $\boldsymbol{\delta\theta}$ is the error in orientation, expressed as an axis-angle vector. Furthermore, $\boldsymbol{\delta a_b}$ and $\boldsymbol{\delta\omega_b}$ are the considered as global errors. The motion model for the prediction step of the filter is given below.

Nominal state update:

$$\check{\mathbf{p}}_k = \hat{\mathbf{p}}_{k-1} + \hat{\mathbf{v}}_{k-1}\Delta t + \frac{1}{2}\left(\mathbf{R}_{inert,body}\left(\mathbf{a}_{m_{k-1}} - \hat{\mathbf{a}}_{b_{k-1}}\right) + \mathbf{g}\right)\Delta t^2 \tag{2.4}$$

$$\check{\mathbf{v}}_k = \hat{\mathbf{v}}_{k-1} + \left(\mathbf{R}_{inert,body}\left(\mathbf{a}_{m_{k-1}} - \hat{\mathbf{a}}_{b_{k-1}}\right) + \mathbf{g}\right)\Delta t \tag{2.5}$$

$$\check{\mathbf{q}}_k = \hat{\mathbf{q}}_{k-1} \otimes \mathbf{q}\left\{\left(\boldsymbol{\omega}_{m_{k-1}} - \hat{\boldsymbol{\omega}}_{b_{k-1}}\right)\Delta t\right\} \tag{2.6}$$

$$\check{\mathbf{a}}_{b_k} = \hat{\mathbf{a}}_{b_{k-1}} \tag{2.7}$$

$$\check{\boldsymbol{\omega}}_{b_k} = \hat{\boldsymbol{\omega}}_{b_{k-1}} \tag{2.8}$$

with

$$
\begin{align}
\mathbf{a}_{m_k} &= \text{Acceleration measured by accelerometer at k}^{\text{th}} \text{ instance} \tag{2.9}\\
\boldsymbol{\omega}_{m_k} &= \text{Angular velocity measured by gyroscope at k}^{\text{th}} \text{ instance} \tag{2.10}\\
\mathbf{R}_{inert,body} &= \text{Rotation matrix corresponding to } \hat{\mathbf{q}}_{k-1} \tag{2.11}\\
\mathbf{g} &= \text{Gravity vector w.r.t inertial frame} \tag{2.12}\\
\otimes &= \text{Quaternion composition operator.} \tag{2.13}
\end{align}
$$

Error state covariance matrix update:

$$
\check{\mathbf{P}}_k = \mathbf{F}_x \hat{\mathbf{P}}_{k-1} \mathbf{F}_x^T + \mathbf{F}_i \mathbf{Q}_i \mathbf{F}_i^T \tag{2.14}
$$

where

$$
\begin{align}
\mathbf{P}_k &= \text{Error state covariance matrix at k}^{\text{th}} \text{ instance} \tag{2.15}\\
\mathbf{F}_x &= \left.\frac{\partial f}{\partial \boldsymbol{\delta}\mathbf{x}}\right|_{\mathbf{x}=\hat{\mathbf{x}}_{k-1}, \boldsymbol{\delta}\mathbf{x}=\mathbf{0}, \mathbf{w}=\mathbf{0}} \tag{2.16}\\
\mathbf{F}_i &= \left.\frac{\partial f}{\partial \mathbf{w}}\right|_{\mathbf{x}=\hat{\mathbf{x}}_{k-1}, \boldsymbol{\delta}\mathbf{x}=\mathbf{0}, \mathbf{w}=\mathbf{0}} \tag{2.17}\\
f(.) &= \text{Function representing the motion model} \tag{2.18}\\
\mathbf{w} &= \text{Process noise vector} \tag{2.19}\\
\mathbf{Q}_i &= \text{Process noise covariance matrix.} \tag{2.20}
\end{align}
$$

The equations pertaining to the correction process, upon receiving a measurement update is given below.

$$
\begin{align}
\mathbf{K}_k &= \check{\mathbf{P}}_k \mathbf{H}^T \left(\mathbf{H} \check{\mathbf{P}}_k \mathbf{H}^T + \mathbf{V} \right)^{-1} \tag{2.21}\\
\hat{\mathbf{P}}_k &= \left(\mathbf{I} - \mathbf{K}_k \mathbf{H} \right) \check{\mathbf{P}}_k \tag{2.22}\\
\hat{\boldsymbol{\delta}\mathbf{x}}_k &= \mathbf{K}_k \left(\mathbf{y} - h\left(\check{\mathbf{x}}_k \right) \right) \tag{2.23}\\
\hat{\mathbf{x}}_k &= \check{\mathbf{x}}_k \oplus \hat{\boldsymbol{\delta}\mathbf{x}}_k \tag{2.24}\\
\hat{\boldsymbol{\delta}\mathbf{x}}_k &\leftarrow \mathbf{0}. \tag{2.25}
\end{align}
$$

Here,

$$h(.) = \text{Measurement function} \tag{2.26}$$

$$\mathbf{H} = \text{Jacobian of the measurement function, relative to the error state, evaluated at zero} \tag{2.27}$$

$$\mathbf{V} = \text{Measurement noise matrix} \tag{2.28}$$

$$\mathbf{y} = \text{Measurement vector} \tag{2.29}$$

$$\mathbf{I} = \text{Identity matrix of suitable dimension} \tag{2.30}$$

$$\oplus = \text{Operator representing the combination of nominal state and error state vectors.} \tag{2.31}$$

For a detailed presentation of the matrices included in above equations, refer Appendix 2.3.5[19].

2.3.2 Stochastic cloning and backward smoothing

Following the works of Emter et. al[20], we have implemented a stochastic cloning framework along with the ES-EKF, as a mean of integrating relative measurements obtained from sensors such as wheel odometer and visual/LiDAR odometry algorithms. Since a relative measurement relates the current state of the vehicle to a previous state, it is required to propagate the corrections resultant from absolute measurements to all the previous states as well. In-order to achieve this, a Rauch-Tung-Striebel (RTS) backward smoother was integrated.

2.3.3 State buffer

It is essential to store a window of previous estimates obtained as the output of the ES-EKF, in-order to be used when fusing a relative measurement, under stochastic cloning framework. A buffer with predefined length was implemented to achieve this functionality. Once a prediction is done by the ES-EKF, the estimate is added to the buffer. If the buffer is full, the oldest estimate will be removed, keeping the buffer length a constant.

Other than the estimate itself, the motion model inputs that resulted the prediction of the estimate and the prediction covariance matrix are also stored in this buffer. In addition to the stochastic cloning mechanism, they are used in propagating the effect of an absolute measurement correction across all the states of the buffer.

2.3.4 Time synchronization

Measurements from different sensors arrive at different times, in different rates. These asynchronous arrivals are handled using the ROS's in-built multi-threading behaviour of subscriber callback functions. Each sensor publishes data to its own ROS topic, for which the localization module is subscribed to. Hence, each publication will invoke a callback function, which includes the logic for carrying out either a prediction or a correction step, depending on the measurement received.

When an absolute correction is to be carried out, the filter first searches the state buffer, for the state with the closest timestamp to that of the measurement. Then it performs correction steps on that state. After it has been completed, the effect of the correction has to be propagated to all the remaining states in the buffer. States having later timestamps will be adjusted by re-predicting them based on the corrected state. States with earlier timestamps will be adjusted through performing RTS smoothing. A similar procedure is carried out upon receiving a relative measurement, except that, backward smoothing with RTS will not be performed.

In this manner, a delayed measurement can still contribute to a correction, unless it is too delayed, so that the state corresponding to its timestamp no longer exists in the state buffer.

If a prediction is to be carried out, the filter will only search for the latest timestamp of the states. If it is greater than that of the prediction input, the input will be neglected. Otherwise, a prediction will be carried out and the newly predicted state will be added to the state buffer.

2.3.5 Zero Velocity Update measurements

Some of the physical constraints that govern the motion of a vehicle can be incorporated into the sensor fusion mechanism, in-order to make the estimate more accurate. Zero Velocity Update (ZUPT) measurements, generated based on the fact that the motion of a vehicle is constrained only towards its longitudinal direction, are one such constraint that can be employed to couple the positional measurements with the heading estimation [21].

However, it should be noted that, the advantages of ZUPT measurements come at a cost. Since they are fed into the ES-EKF as ordinary absolute measurements, they reduce the estimated covariance of the error, causing the filter to be overconfident on its estimate. This effect should be balanced out by carefully tuning the measurement variance of the ZUPT measurement. Furthermore, once diverged due to erroneous measurements, the estimate takes a longer time to converge back to the ground truth, after starting to receive correct measurements.

REFERENCES

[1] S. Liu, J. Tang, Z. Zhang, and J. Gaudiot, “Computer architectures for autonomous driving,” *Computer*, vol. 50, no. 8, pp. 18–25, 2017.

[2] A. C. Hernandes, A. S. Brito, H. Roncancio, D. V. Magalhães, M. Becker, R. C. B. Sampaio, and B. T. Jensen, “Gisa: A brazilian platform for autonomous cars trials,” in *2013 IEEE International Conference on Industrial Technology (ICIT)*, 2013, pp. 82–87.

[3] P. Lamon, S. Kolski, R. Triebel, R. Siegwart, and W. Burgard, “The smartter for elrob 2006 - a vehicle for fully autonomous navigation and mapping in outdoor environments,” *Navigation*, 2006.

[4] M. Montemerlo, J. Becker, S. Bhat, H. Dahlkamp, D. Dolgov, S. Ettinger, D. Haehnel, T. Hilden, G. Hoffmann, B. Huhnke, D. Johnston, S. Klumpp, D. Langer, A. Levandowski, J. Levinson, J. Marcil, D. Orenstein, J. Paefgen, I. Penny, A. Petrovskaya, M. Pflueger, G. Stanek, D. Stavens, A. Vogt, and S. Thrun, “Junior: The stanford entry in the urban challenge,” *Journal of Field Robotics*, vol. 25, no. 9, pp. 569–597, 2008. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/rob.20258

[5] “Ieee spectrum: Technology, engineering, and science news, 2020.” [Online]. Available: http://spectrum.ieee.org/automaton/robotics/artificialintelligence/googles-autonomous-car-takes-to-the-streets

[6] M. Bertozzi, A. Broggi, E. Cardarelli, R. I. Fedriga, L. Mazzei, and P. P. Porta, “Viac expedition toward autonomous mobility [from the field],” *IEEE Robotics & Automation Magazine*, vol. 18, no. 3, pp. 120–124, 2011.

[7] “Apollo open platform.” [Online]. Available: https://apollo.auto/developer.html

[8] “Autoware.auto.” [Online]. Available: https://www.autoware.auto/

[9] G. Wan, X. Yang, R. Cai, H. Li, Y. Zhou, H. Wang, and S. Song, “Robust and precise vehicle localization based on multi-sensor fusion in diverse city scenes,” in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 4670–4677.

[10] A. Soloviev, “Tight coupling of gps, laser scanner, and inertial measurements for navigation in urban environments,” in *2008 IEEE/ION Position, Location and Navigation Symposium*, 2008, pp. 511–525.

[11] J. Levinson and S. Thrun, “Robust vehicle localization in urban environments using probabilistic maps,” in *2010 IEEE International Conference on Robotics and Automation*, 2010, pp. 4372–4378.

[12] S. I. Roumeliotis, G. S. Sukhatme, and G. A. Bekey, "Circumventing dynamic modeling: evaluation of the error-state kalman filter applied to mobile robot localization," in *Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No.99CH36288C)*, vol. 2, 1999, pp. 1656–1663 vol.2.

[13] J. Levinson, J. Askeland, J. Becker, J. Dolson, D. Held, S. Kammel, J. Z. Kolter, D. Langer, O. Pink, V. Pratt, M. Sokolsky, G. Stanek, D. Stavens, A. Teichman, M. Werling, and S. Thrun, "Towards fully autonomous driving: Systems and algorithms," in *2011 IEEE Intelligent Vehicles Symposium (IV)*, 2011, pp. 163–168.

[14] A. SOLOVIEV, D. BATES, and F. VAN GRAAS, "Tight coupling of laser scanner and inertial measurements for a fully autonomous relative navigation solution," *NAVIGATION*, vol. 54, no. 3, pp. 189–205, 2007. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/j.2161-4296.2007.tb00404.x

[15] I. Baldwin and P. Newman, "Road vehicle localization with 2d push-broom lidar and 3d priors," in *2012 IEEE International Conference on Robotics and Automation*, 2012, pp. 2611–2617.

[16] E. Ward and J. Folkesson, "Vehicle localization with low cost radar sensors," in *2016 IEEE Intelligent Vehicles Symposium (IV)*, 2016, pp. 864–870.

[17] K. Yoneda, H. Tehrani, T. Ogawa, N. Hukuyama, and S. Mita, "Lidar scan feature for localization with highly precise 3-d map," in *2014 IEEE Intelligent Vehicles Symposium Proceedings*, 2014, pp. 1345–1350.

[18] S. P. Won, W. W. Melek, and F. Golnaraghi, "A kalman/particle filter-based position and orientation estimation method using a position sensor/inertial measurement unit hybrid system," *IEEE Transactions on Industrial Electronics*, vol. 57, no. 5, pp. 1787–1798, 2010.

[19] J. Solà, "Quaternion kinematics for the error-state kalman filter," 2017.

[20] T. Emter and J. Petereit, "Stochastic cloning and smoothing for fusion of multiple relative and absolute measurements for localization and mapping," in *2018 15th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, 2018, pp. 1508–1513.

[21] G. Dissanayake, S. Sukkarieh, E. Nebot, and H. Durrant-Whyte, "The aiding of a low-cost strapdown inertial measurement unit using vehicle model constraints for land vehicle applications," *IEEE Transactions on Robotics and Automation*, vol. 17, no. 5, pp. 731–747, 2001.

APPENDIX I