



Master Thesis Proposal

# Factor Graph Based Radar Inertial Odometry

*Ajay Ragh*

Supervised by

PD Dr. rer. nat. Lasse Klingbeil  
M.Sc. Fahmi Rouatbi  
M.Sc. Felix Esser

November 2025

## 1 Introduction

For any robotic system navigating through the real world, it needs to answer a few questions. "What is the environment around it?" and "Where is it in this environment?". Based on the answers to these questions the system can then develop a plan to move through the environment. SLAM (**S**imultaneous **L**ocalization **A**nd **M**apping) is the mainstream approach that is used to solve this challenge by, as the name suggests, simultaneously localizing the system in a map of the environment that is being estimated simultaneously [1]. Sensors like camera, LIDAR (**L**ight **D**etection **A**nd **R**anging) are the most common means of sensing the environment. Coupled with sensors like IMU (**I**ntertial **M**easurement **U**nit) and GPS (**G**lobal **P**ositioning **S**ystem), we are able to localize the system in the environment[1]. But what can be done when the system is being used in a GPS denied environment? What if the sensors like Camera/LIDAR face challenges in reliably sensing the environment?

Weather conditions like heavy rain, snowfall, dense fog or other scenarios that emulate similar environmental conditions pose a challenge for reliable sensing to sensors like camera and LIDAR[2]. Snow can create false detections in LIDAR and occlude the camera view, rain drops can distort the camera view and scatter laser beams leading to false detections in LIDAR. As a result, techniques that rely on these sensors degrades or fails to produce reliable measurements like odometry of the system.

This project aims to solve this particular challenge using RADAR (**R**adio **D**etection **A**nd **R**anging) sensors. RADAR sensors use radiowaves for obstacle detection and range measurements[3]. By virtue of the wavelengths at which these sensor signals operate, these sensors are not susceptible to the challenges caused by snow, smoke, fog, rain etc like camera or LIDAR sensors. We aim to create a basic system using a combination of RADAR sensors and IMU to produce odometry measurement of the system. Currently available 4D mmWave radar sensors can provide us with measurements as pointclouds along with point level measurement of doppler velocity and intensity or RCS (**R**adar **C**ross **S**ection). In case of indoor scenarios, due to the presence of multipath reflections traditional point matching based techniques are challenging to be used with RADAR sensors[4]. Our approach is to use RADAR pointclouds from one or more sensors and then estimate the ego motion of the system by making use of a factor graph[5] framework which combines estimated ego velocity from RADARs and IMU odometry. Additionally we investigate incorporating additional sensors like altimeter or barometer to improve the odometry estimation[6].

## 2 Related Work

There are various types of radar sensors currently available in the market: single beam radars, scanning radars and single chip mmWave radar sensors[4][7][8]. Each of these radars sense the environment in slightly different ways but at it's core, the methodology remains the same. A radio wave is sent from the sensor which on hitting an obstacle gets reflected back. The reflected signal is the received by the sensor which is then processed to obtain information about the obstacle like doppler velocity, received signal power gives information regarding the radar cross section of the object, its position in space measured as range, azimuth and elevation(in case of latest 4D radar antenna arrays[9]).

Navigation using radar sensors has become widely researched in the recent time periods, especially in the automobile industry. After the development of 4D mmWave antenna arrays it has become possible to obtain 3D measurement from radar sensors. These sen-

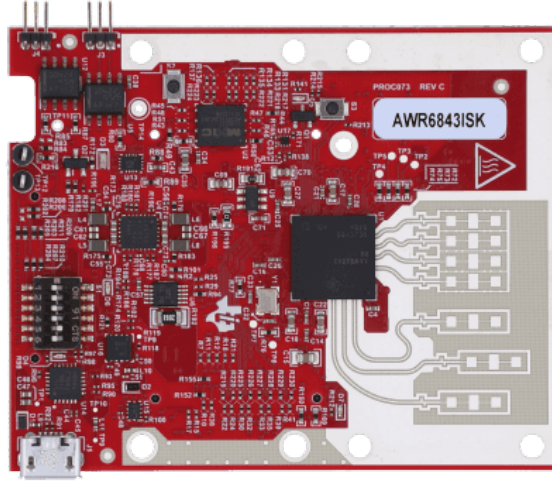


Figure 1: Image of an AWR6843ISK radar array SoC board from Texas Instrumentation

sors can be available as light weight chips as seen in Figure 1, which can be attached on systems like UAS (**U**n**m**anned **A**ircraft **S**ystems). Radar odometry using these sensors can be broadly classified into approaches that perform scan matching, perform instantaneous velocity estimation or hybrid techniques that combine both these together.

**Scan matching** based techniques are mostly techniques that are adapted from approaches used with LIDAR pointclouds. ICP (**I**terative **C**losest **P**oint) techniques[10] and its variations like KISS-ICP[11] are widely used techniques which work robustly on LIDAR pointclouds. In these approaches, consecutive scans are matched to estimate ego state changes between them[12][13][14]. RADAR pointclouds are prone to multipath errors especially in indoor scenarios, outlier point detections etc. Additionally, compared to LIDAR pointclouds, these pointclouds are very sparse and as a result point matching techniques used for LIDAR data cannot provide robust results. There are recent works that try to account for this sparsity by aggregating RADAR scans and also taking into account doppler velocity information provided by RADAR sensors for scan matching[15].

**Instantaneous velocity** estimation techniques makes use of the angle of arrival and doppler velocity information from a single scan to estimate the ego velocity[16][17][18]. The basic operating principle behind these techniques is to first filter out outliers and dynamic point detections in the pointcloud using techniques like RANSAC (**R**andom **S**ample **C**onsensus) algorithm. Once we have only static points, the measured doppler velocity is basically a function of ego velocity and the angle of arrival between the radar sensor and the obstacle. Now, provided we have more than 3 such doppler velocity measurements, the ego velocity can be estimated using a Least Squares Estimation technique[19][20].

**Hybrid techniques** uses a combination of scan matching or similar improved matching techniques[21] to estimate the pose changes between RADAR scans, instantaneously estimated ego velocity[4], inertial sensor measurements[22], or other external sensor data and then fuse this information via optimization techniques or filters like EKF(**E**xtended **K**alman **F**ilter) or Factor Graphs[23][24] to obtain robust ego odometry measurements.

### 3 Problem Statement

The proposed project aims to develop a hybrid approach towards ego odometry estimation using a combination of RADAR pointclouds and IMU raw measurements. We follow the instantaneous ego velocity estimation technique followed in works [17][18] to obtain an estimated ego velocity. Then, we use the IMU raw measurements between two consecutive ego velocity estimation frames to perform IMU preintegration[25]. This provides us with change in state between the two frames. Both the ego velocity estimation and IMU state changes are then fused together using a Factor Graph framework[5]. Additionally, since IMU preintegration results have higher errors in yaw estimation and height estimation, we investigate addition of barometer/altimeters as a source of height information and using scan matching techniques adapted for RADAR pointclouds to obtain a yaw estimate between the frames.

The factor graph will be implemented using GTSAM[26] which allows us to create the custom factors required for velocity, yaw, altitude measurements. The Figure 2 illustrate the structure of the factor graph that we plan to implement. The optimization of the graph will be done using ISAM2[27] which is the current standard algorithm used for factor graph optimization when we are dealing with online scenarios.

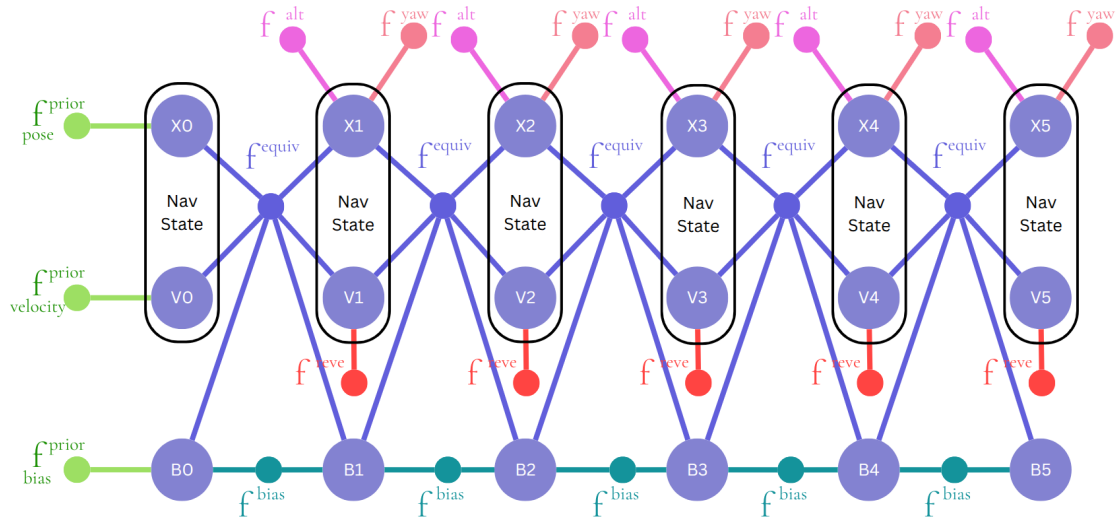


Figure 2: Structure of the factor graph.  $X_i$  denote Pose,  $V_i$  denote Velocity and  $B_i$  denote IMU bias variables respectively.  $f$  denotes the factor nodes described by the corresponding labels. In GTSAM[26] which we use to implement the factor graph we can combine Pose and Velocity into a single variable node called as Navigation State

We will develop our own hardware setup as seen in Figure 3 using off the shelf SoC radar sensor chips and IMU sensors. Once the software implementation of the algorithm explained previously is completed we will evaluate the algorithm against groundtruth odometry available in publicly available datasets like ColoRadar[28] which contains RADAR data collected using similar SoC RADAR chips in indoor and outdoor environments. Additionally we plan to collect our own data using our hardware along with groundtruth data to perform evaluation.

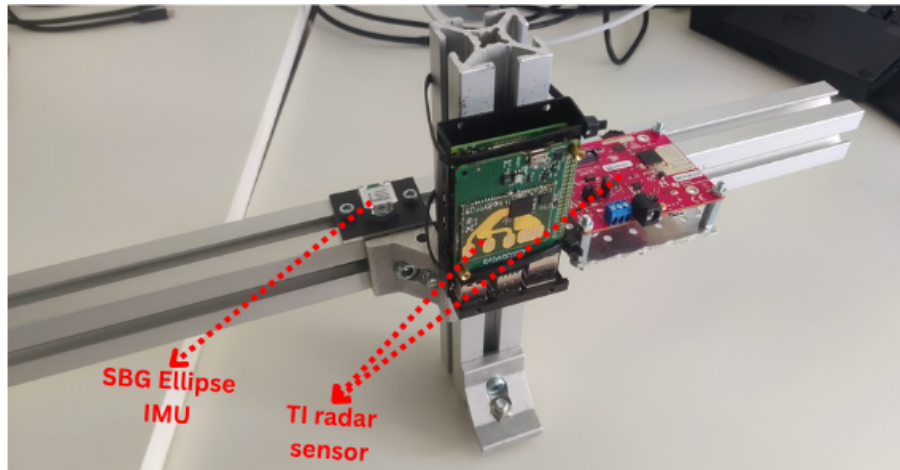


Figure 3: Hardware setup consisting of two TI SoC radar sensor boards and an SBG Ellipse IMU

## 4 Project Plan

### 4.1 Work Packages

#### WP1 Literature Review

- T1.1 Gather literature on radar-inertial odometry techniques, factor graphs, sensor fusion approaches.
- T1.2 Investigate instantaneous velocity estimation techniques, scan matching approaches adapted for radar data and IMU preintegration.
- T1.3 Investigate graph optimization techniques.

#### WP2 Code Implementation

- T2.1 Setup Radar Ego Velocity Estimator code.
- T2.2 Setup scan matching based yaw estimation code.
- T2.3 Setup GTSAM and implemet the factor graph.
- T2.4 Define error calculation logic for Altitude, Velocity and Yaw angle measurements.
- T2.5 Create the custom factors based on the defined error calculation logic.
- T2.6 Debug and test implemtation with datasets and online ROS data.

#### WP3 Evaluation with Public Datasets

- T3.1 Identify and define evaluation metrics.
- T3.2 Evaluate with public datasets like ColoRadar[28].
- T3.3 Visualize and analyze the evaluation results.

#### WP4 Evaluation with own data

- T4.1 Collect data with own hardware indoors and outdoors along with groundtruth data.

T4.2 Perform evaluation based on the error metrics.

T4.3 Visualize and analyze the evaluation results.

### **WP5 Project Report**

T5.1 Create initial draft.

T5.2 Collect supervisor reviews.

T5.3 Refine the report.

T5.4 Final report submission

## **4.2 Deliverables**

### **Minimum Viable**

- Literature review completed.
- Setup proposed factor graph implementation with radar ego velocity and imu preintegration factors.
- Evaluate implementation with public datasets.
- Final draft of the report.

### **Expected**

- All minimum viable deliverables.
- Implement custom factors for altimeter and yaw measurement.

### **Desired**

- All expected deliverables.
- Data collection with own hardware setup including groundtruth.
- Evaluation with collected groundtruth.

## **References**

- [1] L. Carlone, A. Kim, T. Barfoot, D. Cremers, and F. Dellaert, eds., *SLAM Handbook. From Localization and Mapping to Spatial Intelligence*. Cambridge University Press, 2026.
- [2] A. A. Filgueira, H. González-Jorge, S. Lagüela, L. Díaz-Vilariño, and P. Arias, “Quantifying the influence of rain in lidar performance,” *Measurement*, vol. 95, pp. 143–148, 2017.
- [3] M. I. Skolnik, “Introduction to radar,” *Radar handbook*, vol. 2, p. 21, 1962.
- [4] C. Doer and G. Trommer, “x-rio: Radar inertial odometry with multiple radar sensors and yaw aiding,” *Gyroscopy and Navigation*, vol. 12, pp. 329 – 339, 2021.

- [5] D. Koller and N. Friedman, *Probabilistic graphical models: principles and techniques*. MIT press, 2009.
- [6] C. Doer and G. F. Trommer, “Radar inertial odometry with online calibration,” in *2020 European Navigation Conference (ENC)*, pp. 1–10, 2020.
- [7] J. Hasch, E. Topak, R. Schnabel, T. Zwick, R. Weigel, and C. Waldschmidt, “Millimeter-wave technology for automotive radar sensors in the 77 ghz frequency band,” *IEEE Transactions on Microwave Theory and Techniques*, vol. 60, no. 3, pp. 845–860, 2012.
- [8] J. Steinbaeck, C. Steger, G. Holweg, and N. Druml, “Next generation radar sensors in automotive sensor fusion systems,” in *2017 Sensor Data Fusion: Trends, Solutions, Applications (SDF)*, pp. 1–6, 2017.
- [9] M. Stolz, M. Wolf, F. Meinl, M. Kunert, and W. Menzel, “A new antenna array and signal processing concept for an automotive 4d radar,” in *2018 15th European Radar Conference (EuRAD)*, pp. 63–66, 2018.
- [10] A. Segal, D. Haehnel, and S. Thrun, “Generalized-icp,” in *Robotics: science and systems*, vol. 2, p. 435, Seattle, WA, 2009.
- [11] I. Vizzo, T. Guadagnino, B. Mersch, L. Wiesmann, J. Behley, and C. Stachniss, “Kiss-icp: In defense of point-to-point icp—simple, accurate, and robust registration if done the right way,” *IEEE Robotics and Automation Letters*, vol. 8, no. 2, pp. 1029–1036, 2023.
- [12] R. Aldera, D. De Martini, M. Gadd, and P. Newman, “Fast radar motion estimation with a learnt focus of attention using weak supervision,” in *2019 international conference on robotics and automation (ICRA)*, pp. 1190–1196, IEEE, 2019.
- [13] Y. Almalioglu, M. Turan, C. X. Lu, N. Trigoni, and A. Markham, “Milli-rio: Ego-motion estimation with low-cost millimetre-wave radar,” *IEEE Sensors Journal*, vol. 21, no. 3, pp. 3314–3323, 2020.
- [14] M. Rapp, M. Barjenbruch, K. Dietmayer, M. Hahn, and J. Dickmann, “A fast probabilistic ego-motion estimation framework for radar,” in *2015 European Conference on Mobile Robots (ECMR)*, pp. 1–6, IEEE, 2015.
- [15] D. C. Herraiez, M. Zeller, L. Chang, I. Vizzo, M. Heidingsfeld, and C. Stachniss, “Radar-only odometry and mapping for autonomous vehicles,” in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 10275–10282, 2024.
- [16] A. Kramer, C. Stahoviak, A. Santamaria-Navarro, A.-a. Agha-mohammadi, and C. Heckman, “Radar-inertial ego-velocity estimation for visually degraded environments,” in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5739–5746, 2020.
- [17] C. Doer and G. F. Trommer, “An ekf based approach to radar inertial odometry,” in *2020 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, pp. 152–159, 2020.
- [18] D. Kellner, M. Barjenbruch, J. Klappstein, J. Dickmann, and K. Dietmayer, “Instantaneous ego-motion estimation using doppler radar,” in *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*, pp. 869–874, 2013.

- [19] V.-J. Štironja, L. Petrović, J. Peršić, I. Marković, and I. Petrović, “Rave: A framework for radar ego-velocity estimation,” in *2024 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, pp. 1–6, IEEE, 2024.
- [20] H. V. Do, B. S. Ko, Y. H. Kim, and J. W. Song, “Creve: An acceleration-based constraint approach for robust radar ego-velocity estimation,” *arXiv preprint arXiv:2409.16847*, 2024.
- [21] J. Xiang, X. He, Z. Chen, L. Zhang, X. Luo, and J. Mao, “Vgc-rio: A tightly integrated radar-inertial odometry with spatial weighted doppler velocity and local geometric constrained rcs histograms,” *IEEE Robotics and Automation Letters*, vol. 10, pp. 11642–11649, 2025.
- [22] H. V. Do, Y. H. Kim, J. H. Lee, M. H. Lee, and J. W. Song, “Dero: Dead reckoning based on radar odometry with accelerometers aided for robot localization,” in *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 8547–8554, IEEE, 2024.
- [23] W. Yang, H. Jang, and A. Kim, “Ground-optimized 4d radar-inertial odometry via continuous velocity integration using gaussian process,” *arXiv preprint arXiv:2502.08093*, 2025.
- [24] V.-J. Štironja, L. Petrović, J. Peršić, I. Marković, and I. Petrović, “Impact of temporal delay on radar-inertial odometry,” *arXiv preprint arXiv:2503.02509*, 2025.
- [25] C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza, “Imu preintegration on manifold for efficient visual-inertial maximum-a-posteriori estimation,” 2015.
- [26] F. Dellaert, “Factor graphs and gtsam: A hands-on introduction,” *Georgia Institute of Technology, Tech. Rep*, vol. 2, no. 4, 2012.
- [27] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. J. Leonard, and F. Dellaert, “isam2: Incremental smoothing and mapping using the bayes tree,” *The International Journal of Robotics Research*, vol. 31, no. 2, pp. 216–235, 2012.
- [28] A. Kramer, K. Harlow, C. Williams, and C. Heckman, “Coloradar: The direct 3d millimeter wave radar dataset,” *The International Journal of Robotics Research*, vol. 41, no. 4, pp. 351–360, 2022.