

# Radar odometry using mmWave technology for SLAM applications.

Leveraging mmWave Radar Sensor Technology for odometry estimation.

Luis Fernando Rodriguez Gutierrez

*Fachhochschule Dortmund*

*M.Eng. Embedded Systems Engineering*

[luis.rodriguez001@stud.fh-dortmund.de](mailto:luis.rodriguez001@stud.fh-dortmund.de)

**Abstract**—Radar technologies have recently emerged as a promising alternative to traditional odometry methods, which often rely on cameras, LiDAR, wheel encoders, inertial sensors, or GPS. Unlike these modalities, millimeter-wave (mmWave) radar offers robustness under adverse conditions such as fog, rain, or low light, while directly providing both range and Doppler velocity measurements. Previous work has demonstrated that radar can support odometry through global iterative closest point (ICP) alignment of point clouds. Building on this foundation, this work focuses on ego-motion estimation using a dual mmWave radar system complemented by an inertial sensor. The dual-radar configuration is designed to increase spatial coverage and reduce ambiguity in radar detections, addressing a key limitation of single-radar odometry approaches.

Experimental evaluations were conducted in both enclosed environments (laboratory) and open environments (university parking lot). Results show that the proposed framework provides consistent and reliable ego-motion estimation, highlighting the potential of mmWave radar as a cost-efficient substitute or complement to conventional odometry sources in autonomous navigation.

**Index Terms**—Radar Odometry, mmWave Radar, Dual-Radar Configuration, Clustering, ICP, Doppler Velocity, Submap Aggregation, Ego-Motion Estimation

## I. INTRODUCTION

Accurate and reliable ego-motion estimation is a fundamental requirement for mobile robotic systems and autonomous vehicle solutions. It is the basis for localization, mapping, and navigation, and errors in this stage directly affect the overall performance of autonomous systems.

Traditionally, odometry has been estimated using a combination of wheel encoders, inertial measurement units (IMUs), and GPS. In more recent years, cameras and LiDAR have been widely used because they provide dense information about the environment, enabling precise feature extraction and recognition. However, these vision-based and LiDAR-based methods have significant drawbacks. Cameras are highly sensitive to illumination changes. LiDAR systems are costly, and their performance can degrade in adverse weather conditions such as fog, rain, or snow. Both methods also require high computational and memory resources. These limitations create the need for complementary sensing solutions that remain reliable under real-world conditions.

Millimeter-wave (mmWave) radar has emerged as a strong candidate to address these issues. Radar is compact, cost-efficient, and inherently robust to poor lighting and weather. A key advantage of radar is its ability to directly measure Doppler velocity. Unlike cameras or LiDAR, which require frame-to-frame comparisons to estimate motion, radar provides direct measurements of radial velocity. This not only indicates how fast the vehicle is approaching or moving away from an object, but also makes it possible to separate static structures from moving objects and to detect relative motion trends. These Doppler-based measurements provide additional constraints for ego-motion estimation, making radar a unique and valuable sensing modality.

Despite these advantages, radar data also presents challenges. The resulting point clouds are sparse and noisy, and they often include significant amounts of clutter. Radar also has lower angular resolution compared to LiDAR or cameras. These limitations make it difficult to directly apply traditional scan-matching techniques, which are usually designed for dense LiDAR point clouds. Previous work has shown that radar-only odometry and multimodal fusion can improve robustness, but challenges remain when dealing with sparsity, clutter, and the stability of scan registration.

This work investigates the use of mmWave radar sensors mounted on a Ninebot Go-Kart test platform. The system also integrates an IMU and an embedded processing unit. A visual overview of the setup, including sensor placement, is shown in Figure 1. This figure highlights the main components without going into detailed descriptions, which are provided later in the hardware description section.



Fig. 1: Ninebot test-vehicle system.

The contributions of this work can be summarized as follows:

- 1) A radar ego-motion pipeline using mmWave sensors and an IMU for rotation, minimizing hardware cost and system complexity.
- 2) Integration of Doppler velocity and RANSAC filtering to improve the separation of static and dynamic objects.
- 3) Submap aggregation to mitigate point cloud sparsity and improve alignment stability.
- 4) Object tracking via clusters to identify and filter dynamic objects from the ego-motion estimation.
- 5) Experimental validation using real-world data collected from a vehicle-mounted mmWave radar sensor.

## II. SYSTEM DESIGN AND METHODOLOGY

### A. Research Design

The main objective of this work is the development of a radar-based odometry system that estimates the vehicle's ego-motion using mmWave radar sensors mounted at the front of the vehicle, in combination with an IMU for rotation compensation. The motivation behind this approach is to explore radar as a cost-effective and robust alternative to vision- or LiDAR-based odometry, particularly in conditions where these methods tend to fail. This builds on prior evidence that radar can support instantaneous ego-motion estimation through Doppler velocity cues [13].

The system processes radar point cloud data enriched with range, angle, and Doppler velocity to extract accurate motion estimates for ego-velocity. This enables reconstruction of the vehicle's trajectory and provides valuable input for SLAM applications.

As shown in Figure 2, two mmWave radar sensors are mounted at the front of the vehicle, with overlapping fields of view to improve coverage and reduce ambiguity. This configuration represents the initial validation scenario, chosen deliberately as a simple and controlled setup to test the basic functionality of the dual-radar concept. Starting from this baseline allowed us to validate the pipeline step by step before extending the experiments to more complex environments such as open outdoor areas.

The dual-radar arrangement increases point density and stability compared to single-radar approaches, aligning with multimodal strategies for robust state estimation [14], [15].

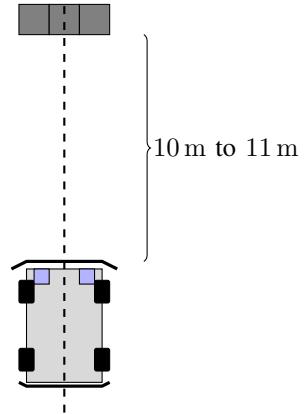


Fig. 2: Test scenario with dual front-mounted mmWave radar sensors. This setup served as the baseline validation environment before moving to more complex scenarios.

By fusing radial speed measurements obtained from Doppler with rotational information from the IMU, the proposed system is designed to remain resilient even in scenarios where LiDAR- or camera-based odometry would degrade. Each radar stream is first processed independently and then merged into a combined representation before feeding into the odometry pipeline.

*1) Sub-Tasks:* The research objective, together with the constraints of using a dual-radar sensor, implied several practical sub-tasks:

- Designing a modular pipeline to acquire and decode synchronized radar data from both sensors.
- Investigating suitable sensor configurations to balance field of view, chirp bandwidth, update rate, and detection density.
- Developing suitable mechanical mounts and selecting optimal sensor placement to ensure stability and maximize coverage.
- Applying RANSAC filtering on Doppler velocities to reject dynamic points and outliers.
- Implementing clustering methods to structure radar detections and isolate relevant features.
- Optimizing the raw radar point cloud using additional information provided by the sensor itself (e.g., SNR, RCS, or range validity) to improve reliability before odometry processing.
- Integrating Doppler velocity information into the odometry estimation process.
- Employing submap aggregation to mitigate sparsity and improve stability.
- Performing ICP alignment between submaps aggregated from both sensors to mitigate sparsity and noise.
- Evaluating the influence of the dual-sensor arrangement on odometry accuracy and robustness.
- Validating the complete system on real-world driving scenarios.

As each sub-task builds upon the results of the previous

one, the work followed an iterative and modular development approach, enabling gradual integration and continuous evaluation of the proposed system. The following sections detail the hardware design, pipeline methodology, and experimental validation.

### B. System Hardware

The hardware setup combines mmWave radars, an inertial sensor, and auxiliary components into a compact test platform. In this section, we describe the role of each element, how they were mechanically integrated, and how the radar configuration was selected for odometry tasks.

#### 1) Sensors:

The sensors that are implemented and used in this project are the following:

##### a) mmWave Radar (IWR6843AOP):

The IWR6843AOPEVM development board from Texas Instruments features the IWR6843AOP, a high performance 4D mmWave FMCW sensor with Antenna On Package (AOP) design. Although IWR6843AOP is intended for industrial applications and its complementary chip, AWR6843AOP, for automotive applications, IWR6843AOP was used in this project because it is available in the form of this development board and the two chips are identical in terms of their functionalities, only differing in compliance with automotive industry [2]. Its small physical size, due to its AOP design, makes it an optimal choice for the desired mounting position, the go-kart's steering column.

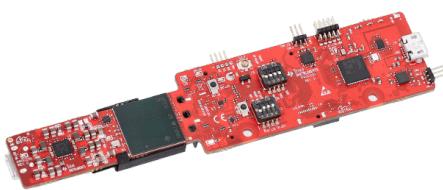


Fig. 3: IWR6843AOP sensor

Source: Texas Instruments, available at [https://www.ti.com/ds\\_dgm/images/fbd\\_swrs219f.gif](https://www.ti.com/ds_dgm/images/fbd_swrs219f.gif)

The IWR6843AOP radar sensor operates within the frequency range of 60 GHz to 64 GHz and integrates 4 receive (RX) and 3 transmit (TX) antennas, radio frequency (RF) front-end stages, analog signal processing, and digital signal processing (DSP). It offers a wide range of communication interfaces including SPI, I2C, CAN-FD, UART and LVDS for raw data access and an Arm Cortex-R4F microcontroller for user-applications as shown in the Figure 4 [3].

As some operating parameters influence each other, their selection must be done carefully while observing the influence of the trade-offs involved. This could be referred to as "sensor tuning" and is a critical step because it directly impacts the system's accuracy and performance.

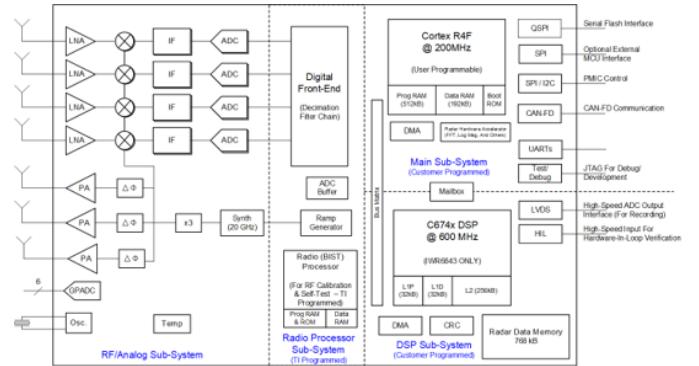


Fig. 4: IWR6843AOP internal block diagram.

Source: Texas Instruments, available at [https://www.ti.com/ds\\_dgm/images/fbd\\_swrs219f.gif](https://www.ti.com/ds_dgm/images/fbd_swrs219f.gif)

The following operating parameters can be tuned:

- Frame rate
- Range resolution
- Maximum unambiguous range
- Maximum radial velocity
- Radial velocity resolution

Tuning these operating parameters introduces trade-offs by influencing each other in the following ways: The resulting

| Tuning Parameter           | Effect on Performance                                | Related HW Block                | Trade-Off                              |
|----------------------------|--|---------------------------------|--|
| Frame Rate                 | Higher FPS = faster updates but more processing load | C674x DSP, Radar Data Memory    | Higher FPS reduces maximum range       |
| Range Resolution           | Higher resolution = better object separation         | ADC, 1D FFT (Range FFT)         | Higher resolution reduces max range    |
| Maximum Range              | Determines farthest detectable object                | RF Front-End, PA, LNA, ADC      | Higher range lowers resolution         |
| Radial Velocity Resolution | Improves speed accuracy                              | DSP, 2D FFT (Doppler FFT)       | Higher resolution requires more chirps |
| Maximum Radial Velocity    | Detects fast-moving objects                          | Chirp rate, TX Antennas, 2D FFT | Higher max velocity reduces resolution |

TABLE I: Radar System Tuning Parameters and Trade-offs

overall accuracy of the velocity and distance measurements is again dependent on these operating parameters:

- **Radial velocity accuracy:** A fine balance between velocity resolution and frame rate must be maintained to ensure precise Doppler shift measurements. Lower resolution results in rounded velocity values, while an excessively high frame rate may introduce computational bottlenecks.
- **Distance accuracy:** Optimizing range resolution and maximum range ensures that detected objects are positioned accurately within the environment. Increasing range often sacrifices resolution, leading to potential inaccuracies in close-range detections.
- **Signal processing considerations:** The FFT calculation parameters directly affect both range and Doppler calculations, influencing the ability to distinguish between objects and detect small velocity variations.

#### b) Inertial Measurement Unit (IMU):

The MTi-G-710 from Xsens is a high-performance GNSS/INS module that integrates a 3D accelerometer, 3D gyroscope, 3D magnetometer, and barometer with an onboard GNSS receiver. Although designed as a complete navigation unit capable of

delivering position and velocity estimates, in this project it was used specifically for its orientation outputs, provided in the form of quaternions, Euler angles, or rotation matrices [19]. Its compact form factor and proven sensor-fusion engine made it a suitable choice for integration into the test platform, complementing the radar subsystem by supplying real-time orientation references.

For the present setup, the device was configured to output quaternions, as this representation avoids singularities inherent in Euler angles and provides numerical stability for continuous tracking. These quaternions were used to track the vehicle attitude and to align radar point clouds within a consistent vehicle-centric coordinate frame. By relying on the MTi-G-710 internal fusion engine rather than implementing custom filtering, the system leveraged factory-calibrated algorithms for bias correction and drift reduction, minimizing processing overhead on the host platform.

To interface with the device and parse its MTData2 protocol, an open-source implementation from Scottapotamas was adopted [20]. This repository enabled robust packet parsing and quaternion extraction in Python, ensuring seamless integration of the IMU data stream into the radar processing pipeline.



Fig. 5: MTi-G710 high-performance inertial navigation system (INS).

Source: *Movella*, available at <https://www.movella.com/sensor-modules/xsens-mti-7-gnss-ins>

### c) Webcam:

A Logitech C270 HD webcam was included in the experimental platform for the sole purpose of video recording. Unlike the radar and IMU, the camera was not integrated into the sensing or processing pipeline and did not contribute any data to the odometry estimation system. Its role was limited to providing visual documentation of the test environment,

allowing experiments to be reviewed and contextualized with synchronized video footage.

The Logitech C270 is a consumer-grade USB webcam capable of capturing video at a resolution of  $1280 \times 720$  pixels and 30 Hz. Its compact design and ease of deployment made it a practical choice for recording without interfering with the radar sensors or mechanical mounts. The resulting video logs were used as qualitative references to better understand the test scenarios and to visually verify environmental conditions during system validation.

*2) Mechanical Integration:* The mechanical design and integration of the sensing suite were crucial to ensure stable operation, accurate data alignment, and reproducibility across experiments. This subsection is divided into three parts:

- 3D printed mounts for flexible radar positioning.
- 3D dedicated IMU mounting plate.
- Geometric placement of radars and corresponding coordinate transformations.

*a) Radar Mounting Design:* A standard mounting bracket is provided by the manufacturer for the IWR6843AOP sensor, as shown in Figure 6a. While suitable for simple setups, this bracket only allows a fixed mounting position without the possibility of adjusting yaw or pitch. Such limitations restrict the ability to optimize the radar field of view for different vehicle placements and test scenarios.

To overcome this, a custom 3D-printed bracket was developed, enabling adjustable tilt and offering greater flexibility in sensor alignment (Figures 6 and 7). This design provides precise control over the radar orientation, which is essential for minimizing ground reflections, extending horizontal coverage, and ensuring that both radars could be consistently aligned with the vehicle's coordinate frame.

The bracket consists of three main components:

- A stable **mounting base** securely attached to the vehicle platform.
- A **sensor frame** that holds the IWR6843AOP in position.
- An adjustable **tightening screw** that locks the tilt angle once aligned.

The mounts were fabricated using PLA, chosen for its ease of printing and adequate stiffness for field use. They were fixed to the vehicle platform with hot glue, a lightweight yet sufficiently rigid method to ensure stability during operation. This approach minimized vibration-induced artifacts and guaranteed repeatable positioning across experiments.

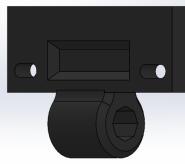
By introducing this adjustable design, the system achieved both mechanical robustness and alignment flexibility, enabling more accurate calibration of the dual-radar setup and improving consistency in downstream data fusion.

While the 3D model (Figures 6-7) illustrates the intended mounting geometry, Figure 8 shows an intermediate real-world implementation. This prototype installation provided a practical check of sensor tilt and positioning but was later replaced by the custom 3D-printed bracket, which ensured stability, repeatability, and precise control over yaw and pitch adjustments.



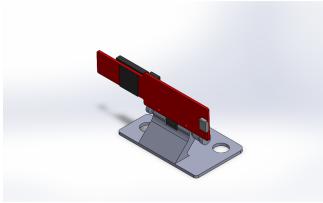
(a) Manufacturer-provided mount for the IWR6843AOP sensor. This design fixes the sensor in place without tilt or yaw adjustment, limiting flexibility.

*Source: amicus engineering, available at amicus.com.sg*

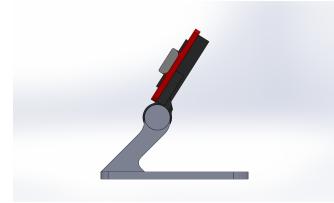


(b) Custom 3D-printed mount designed in this work. It enables controlled tilt adjustments, ensuring repeatable alignment and optimized field of view.

Fig. 6: Comparison of sensor mounting options for the IWR6843AOP radar: (a) fixed manufacturer mount and (b) adjustable 3D-printed design. The custom bracket provides greater stability and alignment flexibility, improving calibration for dual-radar operation.



(a) IWR6843AOP sensor mounted on the custom 3D-printed bracket in its default upright orientation. The mount provides a stable base while allowing tilt adjustments.



(b) Side view showing the tilt functionality of the custom bracket. This adjustability was critical for compensating the 15° upward tilt used in the dual-radar configuration.

Fig. 7: Implementation of the custom 3D-printed mounting solution for the IWR6843AOP radar. The design combines stability with controlled tilt adjustment, enabling repeatable calibration and improved alignment with the vehicle frame.

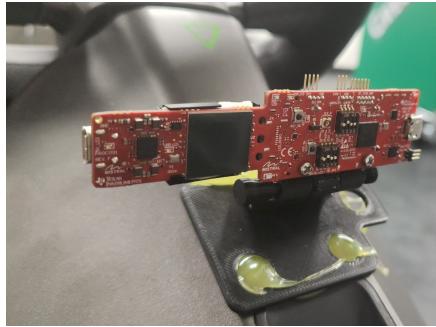


Fig. 8: Prototype installation of the IWR6843AOP radar sensor on the vehicle. This early mounting solution was used for visualization and comparison with the 3D design, but it was not adopted as the final configuration.

*b) IMU Mounting Design:* The MTi-G710 IMU required a stable and horizontally aligned installation point to ensure that its quaternion orientation output could be directly mapped to the vehicle reference frame without introducing additional correction. To achieve this, a custom mounting solution was developed, illustrated in Figure 9.

The 3D design, shown in Figure 9, highlights the three main elements of the assembly:

- 1) The MTi-G710 sensor.
- 2) A flat mounting plate designed to secure the device.
- 3) The rear vehicle mounting spot, located near the spoiler.

This location was selected because it provided a rigid and vibration-resistant base, while also being naturally aligned with the horizontal plane of the vehicle. The design ensured that the IMU could be mounted flush, minimizing installation errors and avoiding unnecessary post-processing of orientation data.

The real-world implementation is shown in Figure 10, where the IMU was securely fixed to the vehicle. This placement offered both ease of integration and the stable reference needed for reliable sensor fusion with the radar system.

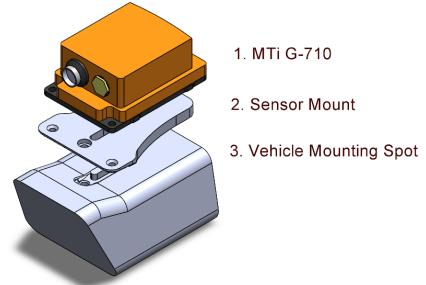


Fig. 9: 3D model of the MTi-G710 IMU installation. The design consists of (1) the IMU, (2) a custom flat sensor mount, and (3) the rear vehicle mounting spot selected for its stability and horizontal alignment.



Fig. 10: Real-world installation of the MTi-G710 at the rear vehicle mount point. This position near the spoiler ensured a rigid and level platform, reducing post-processing requirements for orientation correction.

### 3) Sensor Setup and Calibration:

An understanding of how the sensors operate is required before they can be correctly integrated into the system. This involves studying their configuration tools, calibration procedures, and data output formats. Only by understanding their internal operation and limitations can they be properly adapted for the precise requirements of this application.

## Radar Setup and Calibration

### a) Configuration Tools:

To configure and validate the radar sensors, two official Texas Instruments tools were employed: the *mmWave Demo Visualizer* and the *mmWave Sensing Estimator*. These tools allowed the rapid prototyping of chirp profiles and provided immediate feedback on their impact on range and velocity performance.

The **mmWave Demo Visualizer** [4], [5] is a browser-based application that enables basic sensor configuration and visualization. It allows the user to adjust frame rate, range resolution, maximum range, and velocity settings, while simultaneously observing output plots such as point clouds, noise profile, range profile, and Doppler heatmaps. This tool was particularly useful for testing standard chirp configurations and ensuring the correct operation of the radar hardware in a controlled environment. Figure 11 shows an example of the configuration interface.

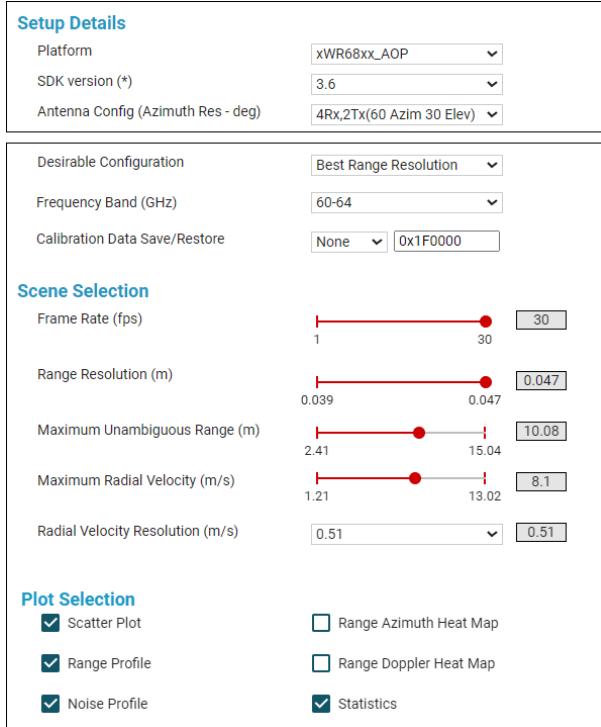


Fig. 11: mmWave Demo Visualizer interface for radar setup and visualization.

Source: Texas Instruments, available at [dev.ti.com](http://dev.ti.com)

Complementing this, the **mmWave Sensing Estimator** [7] provides a theoretical framework to design chirp profiles.

By specifying parameters such as slope, bandwidth, ADC sampling rate, and frame timing, the tool outputs derived radar capabilities including range resolution, maximum detectable velocity, and unambiguous range. However, it must be emphasized that the estimator only generates feasible profiles—it does not guarantee hardware compatibility. Each configuration must be tested on the actual radar device to confirm correct execution. Figure 12 illustrates the chirp design interface of the estimator.

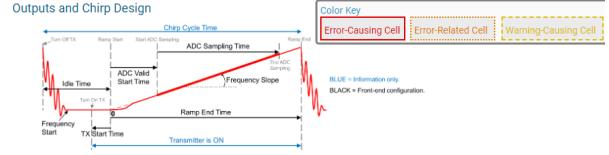


Fig. 12: mmWave Sensing Estimator interface for chirp configuration design.

Source: Texas Instruments, available at [dev.ti.com](http://dev.ti.com)

### b) Sensor Transformations:

Each radar sensor in the dual-radar configuration was mounted with a yaw offset of  $\pm 30^\circ$ , a pitch tilt of  $15^\circ$  upward, and a lateral displacement from the vehicle centerline. To ensure all detections could be expressed in a coherent vehicle-centric frame, the following extrinsic corrections were applied in sequence:

- 1) **Yaw rotation**  $R_{yaw}$ : compensates for the  $\pm 30^\circ$  mounting angle around the Z-axis.
- 2) **Pitch rotation**  $R_{pitch}$ : corrects the  $15^\circ$  upward tilt by rotating around the X-axis.
- 3) **Translation**  $\vec{T}$ : shifts the point clouds by the physical offsets of each radar ( $x = -0.32$  m for left,  $x = -0.28$  m for right).

The transformation from radar to vehicle coordinates is expressed as:

$$\vec{p}_{veh} = R_{yaw} \cdot R_{pitch} \cdot \vec{p}_{radar} + \vec{T} \quad (1)$$

These corrections ensured that detections from both radars overlapped coherently, avoiding duplicated or misaligned clusters. Without them, raw dual-sensor data appeared inconsistent, particularly when fusing Doppler-based velocity estimates. After applying yaw, pitch, and translation adjustments, the fused point cloud served as a stable input for clustering, odometry, and tracking.

### c) Chirp Configuration:

The IWR6843AOP operates as a Frequency-Modulated Continuous Wave (FMCW) radar, transmitting chirps whose frequency increases linearly over a bandwidth  $B$  during a duration  $T_c$ . Reflected signals return with a frequency shift that encodes target range, while Doppler shifts across successive chirps provide relative velocity. The key design parameters include start frequency  $f_0$  [GHz], slope  $S$  [Hz/s], chirp duration  $T_c$  [s], idle time  $T_{idle}$  [s], sampling rate  $f_s$  [Hz], and resulting bandwidth  $B = S \cdot T_c$  [Hz].

From these parameters, the radar capabilities are determined as:

$$\Delta R = \frac{c}{2B} \quad [\text{m}], \quad R_{\max} = \frac{cf_s}{2S} \quad [\text{m}],$$

$$\Delta v = \frac{\lambda}{2N_c T_c} \quad [\text{m/s}], \quad v_{\max} = \frac{\lambda}{4T_c} \quad [\text{m/s}]$$

where  $\Delta R$  is the range resolution,  $R_{\max}$  the maximum unambiguous range,  $\Delta v$  the velocity resolution, and  $v_{\max}$  the maximum unambiguous velocity.

In a dual-radar configuration, chirp design must also address mutual interference. If both devices transmit wideband chirps simultaneously, spectral overlap can generate ghost detections and degraded signal quality. To mitigate this, strategies such as non-overlapping frequency allocations, interleaved chirps, or slope diversity are used. This ensures orthogonality between radars, preserving the integrity of fused point clouds.

For this work, the chirp profiles were derived using the TI Radar Estimator tool, which balances range precision and velocity resolution while ensuring feasible hardware constraints. Wide-band sweeps provided fine range resolution, while multiple chirps per frame ensured adequate Doppler resolution for ego-motion estimation. The resulting configuration supported robust operation in the dual-radar setup without excessive interference.

## IMU Setup and Calibration

### a) Configuration Tools:

The MTi-G-710 was first evaluated with *Xsens MT Manager* to understand the sensor outputs and axis conventions. Using the tool, quaternion orientation was visualized in real time while the unit was held still and under gentle rotations to verify stability and drift. Reference frames were checked so that the IMU heading and gravity vector aligned with the vehicle-centric frame used in the radar pipeline. Output rate, message content, and quaternion format were confirmed in this stage before any integration work. This step served only for validation and sanity checks; the final system consumed the IMU stream directly during experiments.

### b) Calibration Aspects:

The IMU was mounted on a rigid, nearly horizontal plate at the rear of the vehicle to minimize vibrations and to avoid introducing tilt that would require post-processing. On power-up, the device was initialized with a short configuration sequence to enable quaternion output at the desired rate, as specified in the low-level protocol documentation [21]. This sequence places the device in configuration mode, sets the output configuration (quaternions at a fixed sample rate), and returns it to measurement mode for streaming. Because the MTi-G-710 is factory-calibrated and runs an internal sensor-fusion engine, no additional field calibration was required beyond correct mounting and a stable magnetic/metal environment [19]. GNSS and velocity outputs were not used in this work; only quaternions were consumed to provide a reliable attitude reference for aligning radar point clouds.

## III. SUMMARY AND OUTLOOK

Insert conclusion here

## REFERENCES

- [1] Segway Inc., “Ninebot Go-kart PRO product page”, 2025, Webpage. [Online]. Available: [https://de-de\(segway.com/products/ninebot-gokart-pro](https://de-de(segway.com/products/ninebot-gokart-pro)
- [2] Texas Instruments, “IWR1642: difference between AWR and IWR parts”, 2025, Webpage. [Online]. Available: <https://e2e.ti.com/support/sensors-group/sensors/f/sensors-forum/742730/iwr1642-difference-between-awr-and-iwr-parts>
- [3] Texas Instruments, “IWR6843AOPEVM product page”, 2025, Webpage. [Online]. Available: [https://www\(ti.com/tool/IWR6843AOPEVM](https://www(ti.com/tool/IWR6843AOPEVM)
- [4] Texas Instruments, “User’s Guide mmWave Demo Visualizer,” 2020, Online Document. [Online]. Available: [https://www\(ti.com/lit/ug/swru529c/swru529c.pdf?ts=1742817596204](https://www(ti.com/lit/ug/swru529c/swru529c.pdf?ts=1742817596204).
- [5] Texas Instruments, “mmWave Demo Visualizer”, 2025, Webpage. [Online]. Available: [https://dev.ti.com/gallery/view/mmwave/mmWave\\_Demo\\_Visualizer/ver/3.6.0/](https://dev.ti.com/gallery/view/mmwave/mmWave_Demo_Visualizer/ver/3.6.0/)
- [6] Texas Instruments, “Understanding UART Data Output Format”, 2025, Webpage. [Online]. Available: [https://dev.ti.com/tirex/content/radar\\_toolbox\\_2\\_20\\_00\\_05/docs/software\\_guides/Understanding\\_UART\\_Data\\_Output\\_Format.html](https://dev.ti.com/tirex/content/radar_toolbox_2_20_00_05/docs/software_guides/Understanding_UART_Data_Output_Format.html)
- [7] Texas Instruments, “mmWave Sensing Estimator”, 2025, Webpage. [Online]. Available: <https://dev.ti.com/gallery/view/mmwave/mmWaveSensingEstimator/ver/2.5.1/>
- [8] ub4raf, “Ninebot-PROTOCOL”, 2025, GitHub Repository. [Online]. Available: <https://github.com/ub4raf/Ninebot-PROTOCOL>
- [9] -, “Ninebot ES Communicaton Protocol”, 2019, Webpage. [Online]. Available: <https://cloud.scooterhacking.org/release/nbdoc.pdf>
- [10] -, “numpy.polyfit documentation”, 2025, Webpage. [Online]. Available: <https://numpy.org/doc/stable/reference/generated/numpy.polyfit.html>
- [11] C. Önen, A. Pandharipande, G. Joseph, and N. J. Myers, “Occupancy Grid Mapping for Automotive Driving Exploiting Clustered Sparsity,” *IEEE Sensors Journal*, vol. 24, no. 7, pp. 9240-9250, 2024. [Online]. Available: <https://doi.org/10.1109/JSEN.2023.3342463>.
- [12] D. Casado Herraez, M. Zeller, L. Chang, I. Vizzo, M. Heidingsfeld, and C. Stachniss, “Radar-Only Odometry and Mapping for Autonomous Vehicles,” *arXiv preprint*, 2023. [Online]. Available: <https://arxiv.org/abs/2305.12409>
- [13] S. R. Bhatt, B. S. Nadiger, R. Parthasarathy, and H. M. Shetty, “Instantaneous Ego-motion Estimation Using Doppler Radar,” *IEEE Sensors Letters*, vol. 7, no. 5, pp. 1–4, 2023. [Online]. Available: <https://doi.org/10.1109/LSENS.2023.3244030>
- [14] C. E. Beal, T. Williams, J. Pauli, M. Mukadam, and B. Boots, “Robust Off-Road Autonomy Using Multimodal Sensor Fusion,” in *Proc. of the Conference on Robot Learning (CoRL)*, 2023. [Online]. Available: <https://openreview.net/forum?id=kmiZqSgoAt>
- [15] B. Sundaralingam, C. E. Beal, and B. Boots, “Robust High-Speed State Estimation for Off-Road Autonomous Vehicles,” in *Proc. of Robotics: Science and Systems (RSS)*, 2023. [Online]. Available: <https://openreview.net/forum?id=3JpFLY3ihix>
- [16] GeeksforGeeks, *DBSCAN Clustering in ML — Density Based Clustering*, 2023. [Online]. Available: <https://www.geeksforgeeks.org/dbscan-clustering-in-ml-density-based-clustering/>. [Accessed: 19-Mar-2025].
- [17] MathWorks, *Understanding Kalman Filters, Part 3: Optimal State Estimator*, 2017. Available at: <https://la.mathworks.com/videos/understanding-kalman-filters-part-3-optimal-state-estimator--1490710645421.html> (Accessed: March 23, 2025).
- [18] Texas Instruments, *Radar Toolbox – mmWave Sensor Configuration and Demos*, 2024. Available at: [https://dev.ti.com/tirex/explore/node?node=A\\_\\_ADnb17zK9bSRgZqeAxprvQ\\_\\_radar\\_toolbox\\_\\_1AsIXXD\\_\\_2.20.00.05](https://dev.ti.com/tirex/explore/node?node=A__ADnb17zK9bSRgZqeAxprvQ__radar_toolbox__1AsIXXD__2.20.00.05) (Accessed: March 23, 2025).
- [19] Movella (Xsens), *MTi User Manual*, 2023. [Online]. Available: [https://www.xsens.com/hubfs/Downloads/usermanual/MTi\\_usermanual.pdf](https://www.xsens.com/hubfs/Downloads/usermanual/MTi_usermanual.pdf). [Accessed: 26-Sep-2025].
- [20] Scottapotamas, *Xsens MTi Device Interface and Parser*, GitHub repository, 2020. [Online]. Available: <https://github.com/Scottapotamas/xsens-mti>. [Accessed: 26-Sep-2025].
- [21] Movella (Xsens), *Low-Level Communication Protocol Documentation*, 2023. [Online]. Available: [https://www.xsens.com/hubfs/Downloads/Manuals/MT\\_Low-Level\\_Documentation.pdf](https://www.xsens.com/hubfs/Downloads/Manuals/MT_Low-Level_Documentation.pdf). [Accessed: 27-Sep-2025].