ARGUS

Autonomous Radiation-Guided Unified SLAM

A Mobile Robot for Mapping Hazardous, Access-Restricted Interiors

---

Public Report

Redacted for Confidentiality

Navya Sharma

An eight-week research placement, 2026

Host institution, site, and supervising personnel **withheld**

SANITISED • REDACTED • NOT THE FULL REPORT

This is a deliberately sanitised, redacted version of an internal report, prepared for public viewing. The host institution, personnel, quantitative results, the instrument communication protocol, the reproduction appendices, and facility imagery have been **removed for reasons of confidentiality**. Black bars ([redacted]) mark where specific information has been withheld. It is a summary of the *approach* only — not a complete account, a technical manual, or a data release.

Want more detail, or the full report?
Substantial additional material exists but is not published here.
To request further information or the complete report, please contact the author:

contact@navyasharma.org • github.com/navyasharmabits

Access to withheld material is provided at the author's discretion and subject to confidentiality obligations.

Prepared as a portfolio summary • not a technical manual, reproduction guide, or data release

Confidentiality & Redaction Notice

This document is a public summary of a longer internal engineering report. The work was carried out during a research placement at a national research institution whose **name, sites, personnel, and internal procedures are withheld**. It concerns the robotic mapping of hazardous interiors, and both the operational context and a substantial amount of the technical and measured content are treated as confidential.

The following categories of material have been **removed or blacked out** in this version, and their absence should be assumed throughout:

- **The host institution and all personnel** — names, sites, supervisors, and the operational setting in which the work was performed.
- **All quantitative results and measured data** — field measurements, counts, durations, calibration values, and the tables and figures derived from them.
- **The instrument communication protocol** — the reverse-engineered serial framing, frame structure, unit encoding, and scaling that let the survey instrument be read. This was the single largest piece of original work in the project and is withheld in full.
- **Numeric system parameters** — sensor geometry offsets, detector dimensions, grid resolutions, update rates, and similar figures.
- **Facility-identifying imagery** — in-situ photographs and any measured hazard-field maps.
- **The reproduction appendices** — the kinematic derivation, the decoding specification, the mapper algorithm, and the interface contract, which in the original are sufficient to rebuild the system, are omitted entirely.

What remains is the *engineering narrative* — the problem, the reasoning, the design decisions, and the transferable lessons — conveyed at a level that does not depend on the withheld specifics. Wherever a specific value, name, or method has been removed, a black bar ([redacted]) or a bracketed note marks the omission so that the redaction is visible rather than silent. **This report should not be read as a complete account, and the withheld material cannot be shared.**

Contents

Abstract

This report documents the design, integration, and validation of ARGUS (Autonomous Radiation-Guided Unified SLAM), a mobile robotic platform for mapping the interiors of hazardous and otherwise inaccessible spaces. The platform combines a four-wheel mecanum drive for omnidirectional mobility, a 360° LiDAR and a stereo depth camera for simultaneous localisation and mapping (SLAM), and a hand-held survey instrument integrated as a live sensor. The onboard compute runs a containerised robotics middleware stack, with the software graph exposed over the local network so the operator console can be decoupled from the platform.

The principal technical contribution is the integration of the survey instrument as a first-class, *pose-resolved* sensor. The instrument exposes no documented machine interface; its output was reverse-engineered from first principles so that live, correctly scaled readings could be published continuously. *The details of that work are redacted here ([redacted], [redacted]).* Each reading is resolved to the true position of the detector rather than to the robot's geometric centre — an offset of [redacted] which, if neglected, smears the resulting map on every turn in a manner easily mistaken for sensor noise. Readings accumulate into a purpose-built dual-grid mapper that maintains two independent representations of the same data: a conservative map for human interpretation that never guesses, and a strict cost grid for the navigation planner that must. The separation resolves a genuine conflict between what a scientific record and a path planner each require from the same measurements.

The system was demonstrated end to end: sensing, pose resolution, hazard mapping, 2D SLAM, 3D reconstruction, autonomous frontier exploration, and hazard-aware path planning were each implemented and observed working on the physical platform. A staged, bottom-up validation campaign confirmed the driver's stability over extended sessions, characterised the ambient field, and verified detector response against live sources under supervision. *The quantitative results of that campaign are withheld.* Two findings of general engineering significance emerged and are retained in summary form: that certain subsystems cannot share the platform's compute without one silently starving another, and that a slow, small-area detector cannot clear ground as fast as the platform wishes to traverse it — each of which shaped the architecture in ways discussed herein.

1 Introduction

1.1 Background

Some facility interiors are, from a mapping and inspection standpoint, effectively unknown environments. As-built layouts may not reflect installed equipment; the geometry is often irregular and cluttered with process machinery; and human walkthroughs for verification or planning carry an inherent cost — in the setting this project addresses, a cost that constrains how long and how often a person may enter.

A recurring requirement across such facilities is the ability to generate an accurate, geometrically faithful map of an enclosure's interior without continuous human presence inside it. A more demanding requirement, and the one this project ultimately addresses, is to *annotate* that map with the hazard field measured throughout it — so that the output is not merely a floor plan, but a statement about where in the enclosure it is, and is not, safe to send a person.

> **Redacted:** the specific facility, its operational role, and the nature of the enclosures surveyed are withheld. The description above is kept general on purpose.

The system is named ARGUS — Autonomous Radiation-Guided Unified SLAM — after the many-eyed watchman of Greek myth, a fitting metaphor for a machine that fuses LiDAR, stereo vision, and hazard sensing into a single view of a space.

1.2 Motivation

The first motivation is the reduction of human exposure. Robotic platforms have a long, well-documented history of substituting remote mobility and sensing for human entry in hazardous and post-accident environments — the deployment of tracked and wheeled robots for reconnaissance following the Fukushima Daiichi accident is a widely cited example [17]. This project draws on the same underlying motivation but applies it to the more routine, yet still operationally important, problem of interior mapping for inspection, planning, and the digitisation of layouts that currently exist only as outdated drawings or institutional memory.

The second motivation is technical. Many existing remote-inspection deployments rely on either tele-operated platforms with no autonomy — placing the whole navigational burden on a remote operator — or on a single sensing modality at a time. Demonstrating that a low-cost mobile base with a fused, multi-sensor SLAM pipeline can autonomously and reliably map a cluttered indoor space is a meaningful step toward more capable, less operator-intensive tools.

The third motivation is the one that distinguishes this project from a conventional mapping exercise, and it is worth stating plainly. *The hazard is invisible.* It cannot be inferred from geometry, a camera image, or a LiDAR return; it must be measured, at a position, and recorded there. A robot that maps a contaminated enclosure but does not measure it produces a floor plan of a room nobody has learned anything new about. A robot that measures but cannot say where it was standing produces a number without a referent. The interesting engineering problem is therefore neither mapping nor detection, but the faithful *registration* of the second onto the first — and then the use of that registration to govern the robot's own motion.

1.3 Problem Statement

The project addresses the following problem: *design, build, and validate a remotely operable mobile robotic platform that can autonomously construct a geometrically accurate map of an unknown, cluttered indoor environment using onboard LiDAR and stereo-vision sensing; that integrates a calibrated survey instrument as a live, pose-resolved sensor so as to co-register a measured hazard field onto that map; and that uses the resulting field to inform its own path planning, so that the platform declines to traverse ground it has measured as hazardous or has not measured at all.*

1.4 Objectives

The specific objectives set out for the project were:

1. Build a mobile robotic base capable of omnidirectional movement, suited to navigating cluttered, narrow interiors.
2. Integrate a 360° LiDAR and a stereo depth camera, and use their outputs to produce both a 2D occupancy grid for planning and a dense 3D reconstruction for review.
3. Integrate a hand-held survey instrument as a live sensor, decoding its output into correctly scaled physical readings.
4. Resolve each reading to the true pose of the detector rather than the robot's geometric centre, and accumulate readings into a map co-registered with the SLAM map.
5. Implement hazard-aware path planning, such that the planner treats a measured hot region as an obstacle and declines to route through unmeasured ground.
6. Implement autonomous frontier-based exploration.
7. Deliver an operator interface usable without specialist expertise, and document the system to a handover standard.
8. Validate on a representative mock-up layout, and verify detector response against live sources under supervision.

All eight were implemented and demonstrated on the physical platform, with the one important caveat recorded in Section 5.2.

1.5 Scope and Its Evolution

The shape of the work changed materially over the placement, and the reasons are themselves part of the engineering record. Two changes of direction are worth noting without their specifics.

First, **the survey instrument moved from a pending item to the centre of the project.** When it became available it proved to have no documented machine interface at all, and the reverse-engineering of its output became the single largest piece of original work — work that had not been anticipated, because there was no reason to expect a professional instrument's output to be undocumented. In hindsight this was the correct thing to have spent the time on: it is the part that could not have been bought, borrowed, or looked up. *Its details are redacted from this report.*

Second, **a tethered-power architecture was investigated and then deliberately set aside**, the remaining time reallocated to the hazard-sensing chain. Recording the reasoning behind a descoping decision is as much part of an honest account as the work that was completed.

2 Literature Review

This chapter reviews the general, published literature the project drew on. It contains no confidential material and is retained largely intact, as it situates the design choices that follow.

2.1 Simultaneous Localisation and Mapping

SLAM is the problem of estimating a robot's trajectory while simultaneously building a map of an unknown environment, using only onboard measurements and without external positioning infrastructure. The probabilistic treatment of Thrun, Burgard and Fox [1] remains the standard reference for the recursive Bayesian formulation underlying most modern SLAM, and the two-part tutorial by Durrant-Whyte and Bailey [2, 3] was the primary starting point, particularly its discussion of the trade-offs between filter-based and graph-based formulations.

For 2D LiDAR SLAM specifically, scan-matching approaches based on the Iterative Closest Point algorithm [4] provide frame-to-frame pose estimates by aligning consecutive scans, while pose-graph optimisation [5] corrects accumulated drift by enforcing consistency at loop closures — when the robot revisits a mapped location. Real-time implementations such as that of Hess et al. [6] show this combination can run live on modest compute. The 2D pipeline adopted here uses a well-maintained open-source toolbox implementing precisely this architecture: local scan matching against a running submap, with a global pose graph corrected at loop closure, and a serialised graph that can be reloaded and extended across sessions [7].

2.2 Visual Odometry, Depth Sensing and 3D Reconstruction

Stereo and monocular visual odometry estimate camera motion from sequential frames by tracking and triangulating features; Scaramuzza and Fraundorfer's tutorial [8] was a key reference for the failure modes of vision-only odometry — sensitivity to poor lighting, textureless surfaces, and motion blur, all plausible in industrial interiors. This motivated using LiDAR rather than vision as the primary metric sensor, reserving the depth camera for what it is genuinely better at: dense reconstruction and the detection of obstacles outside the LiDAR's single scan plane. For 3D reconstruction from an RGB-D sensor, RTAB-Map [9] was selected for its appearance-based loop closure, its memory-management for long-term operation, and its ability to export a height-banded occupancy grid from a completed database.

2.3 Mecanum-Wheel Kinematics

The mecanum wheel, with its angled passive rollers, lets a four-wheeled platform translate in any planar direction and rotate in place without changing chassis orientation, by independently controlling each wheel. Taheri et al.'s kinematic model [10] formed the basis of the platform's inverse kinematics, and the broader treatment of mobile-robot kinematics in Siegwart, Nourbakhsh and Scaramuzza [11] informed the choice of a mecanum drive over differential or skid-steer alternatives, given the emphasis on manoeuvring in narrow, cluttered corridors where in-place rotation and lateral motion are operationally valuable.

2.4 Navigation, Costmaps and Autonomous Exploration

The navigation stack is Nav2 [12]. Its architecture matters here for one reason: it is not a monolith but a framework of pluggable planners, controllers, and *costmap layers*. A costmap is a layered structure in which each layer contributes cost into a master grid that the planner searches — obstacles, inflation, and a static map are all simply layers. This is what makes hazard-aware navigation tractable rather than a research problem: a measured field is, structurally, just another source of cost, and the design problem reduces to deciding how a measured level should map onto the planner's cost scale. The spatio-temporal voxel layer [13] is relevant as an alternative for representing obstacles outside a 2D scan plane. For autonomous exploration, the frontier-based approach of Yamauchi [14] supplies the central insight: the boundary between mapped free space and the unknown is exactly where a robot must go to learn something new, so exploration reduces to repeatedly navigating to the nearest such boundary until none remain.

2.5 Radiation Detection and Measurement

Knoll [15] and Tsoulfanidis and Landsberger [16] were the primary references for detector physics, and three points bear directly on the design.

First, the underlying process is a Poisson process. A count rate is not a fixed quantity measured with some error; it is the mean of a distribution the detector samples. Successive readings of an unchanging field differ, and the fractional fluctuation grows as the rate falls. This is not noise to be filtered away — it is the physics — and it has a direct architectural consequence: a per-cell record must retain more than a running mean, because a single reading cannot characterise a field.

Second, detection efficiency is strongly geometry-dependent. Counts fall off with distance from a surface, so a reading is meaningful only together with the standoff distance at which it was taken. This makes the sensor's mounting geometry a measurement parameter rather than a mechanical detail.

Third, the radiation types differ enormously in their range in air. The most penetrating can be measured usefully from tens of centimetres; the least penetrating has a range of only a few centimetres and is stopped by a sheet of paper. A detector fixed at a height chosen for the penetrating case is, for the short-range case, effectively blind — not through any deficiency of the electronics, but because the particles never reach the window. No signal processing recovers this; the only remedy is geometric, and it motivates the future work of Section 5.3.

2.6 Robotics in Hazardous Environments, and Middleware

Robotic deployment in hazardous settings has its longest track record in post-accident response, most notably the tracked and wheeled robots deployed after the 2011 Fukushima Daiichi accident [17]. While this project's setting is far less severe, the engineering lessons translate: the value of hardwired links in RF-restricted or shielded enclosures, robust mobility on irregular floors, and a sensor suite that tolerates dust, moisture, and radiation. General reviews of autonomous-vehicle sensing [18] informed the choice of LiDAR and depth-camera modalities. The software stack is built on the open-source robotics middleware ROS 2 [19, 20], selected for the maturity of its SLAM, navigation, and sensor-driver ecosystem. Its transform library [21] deserves specific mention: it maintains a time-indexed tree of coordinate frames and answers queries of the form *"where was frame A relative to frame B at time t?"* — the capability, and the time-indexing in particular, that makes a correctly registered hazard map possible at all.

3 Methodology and System Design

This chapter describes the system's design and reasoning at a conceptual level. Specific numeric parameters, the instrument protocol, and the reproduction-level detail of the original report are **redacted**; their omissions are marked.

3.1 System Overview

ARGUS is organised as a layered system, in which each layer depends only on those beneath it. The layering is not incidental: it is what allowed each subsystem to be developed and validated in isolation by faking the inputs of the layer below, and it is what makes the staged validation of Chapter 4 meaningful.

6 Autonomy — frontier exploration

5 Planning — navigation, costmap layers incl. **hazard**

4 World model — 2D SLAM · 3D reconstruction · hazard mapper

3 State — time-indexed transform tree, incl. the detector frame

2 Drivers — motors/encoders · LiDAR · depth camera · survey instrument

1 Platform — containerised middleware on a single-board computer

Figure 3.1: The layered architecture, shown at a general level. Each layer consumes only the layer beneath it. The components in red form the hazard-sensing chain — the project's principal contribution: the driver, the transform frame that gives each reading a position, the mapper that accumulates them, and the cost layer through which they reach the planner. Specific component names and parameters are omitted.

The flow can be stated in a paragraph. Wheel encoders provide odometry, a smooth but slowly drifting estimate of motion. The 2D SLAM stage consumes LiDAR scans alongside that odometry and produces the correction that pins the drifting estimate to a globally consistent map. Together these give every reading a place in the world. The instrument driver reads the survey meter at its native rate and publishes a timestamped value; the mapper asks the transform library where the detector was *at the instant of that reading*, and writes the value into two grids — a smooth one for humans and a strict one for the planner. The planner consumes the second grid alongside its LiDAR obstacle layer, so that a hazardous cell is, to it, as impassable as a wall. Frontier exploration sits on top, repeatedly asking the planner to drive to the nearest unexplored boundary until none remain.

3.2 Mobility Platform

A four-wheel mecanum drive was selected over differential, skid-steer, and tracked alternatives primarily for omnidirectional manoeuvrability in tight, irregular spaces. There is a second reason, specific to surveying and not appreciated at the outset: a holonomic base can *strafe* *along a wall while keeping the detector pointed at the surface* — exactly the motion a survey wants, and exactly the motion a differential-drive base cannot produce. The platform's low-level control implements the standard mecanum inverse-kinematics mapping between a commanded body-frame velocity and the four wheel speeds [10]; the SLAM and navigation layers above issue body-frame velocities without reasoning about individual wheels. *The kinematic derivation and the platform's dimensional parameters are redacted (original Appendix, withheld).*

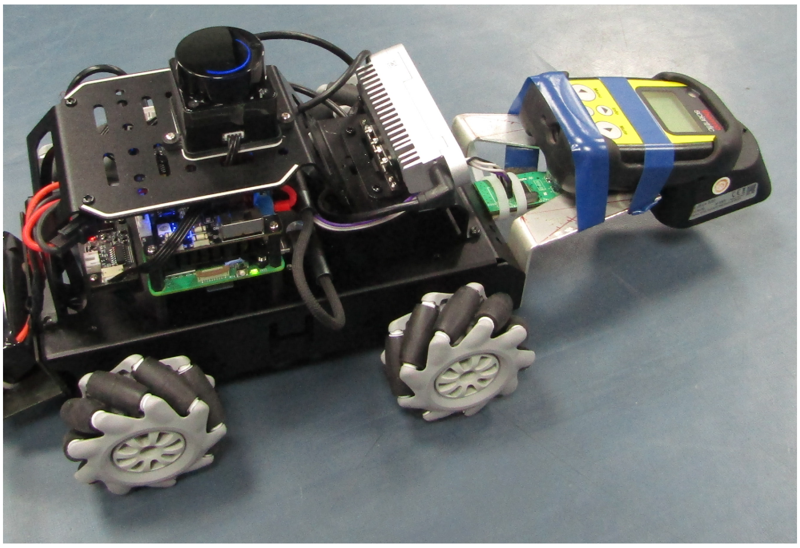


Figure 3.2: The assembled platform. The chassis carries the LiDAR on a central mast and the depth camera at the front; the survey instrument rides on fabricated mounting hardware ahead of and below the chassis — geometry that puts the detector close to the floor and lets it measure ground before the platform drives onto it. Facility-identifying and in-situ imagery is withheld.

3.3 Integrating the Survey Instrument

The survey instrument is a calibrated, professional-grade hand-held device that exposes a machine-readable output — but no documentation of what that output means. There is no published protocol, no vendor SDK, and no existing driver. Its data path therefore had to be characterised from first principles before a single reading could be brought into the system. The decisive insight was general and is safe to state: **an unintelligible serial stream is far more often a framing mismatch than a proprietary binary protocol.** What appeared to be a reverse-engineering problem of indefinite scope resolved, once the framing was correct, into plain readable text.

[The instrument's link parameters, frame structure, unit encoding, checksum, and decimal scaling are redacted in full.]

[redacted]

Two implementation details are worth recording at a general level, because both were arrived at the hard way and both generalise. First, the USB-serial bridge occasionally reports a zero-length read — a momentary glitch as the optical link shifts while the platform moves — which the serial library misreports as a disconnection; a driver that trusts that report suffers phantom disconnections every few minutes on a perfectly healthy link, so the driver here treats a read glitch and a genuine unplugging differently. Second, the driver *refuses to guess* among ambiguous serial ports rather than risk opening the LiDAR's port by mistake and wedging it — failing loudly being preferable to succeeding at the wrong thing.

The rate constraint

The instrument updates at roughly its own fixed rate, which cannot be raised in software. This single figure — *redacted here* — is the constraint around which the entire mapping architecture is designed. Its consequence is arithmetic: a small-area detector reporting at that rate validates at most a small number of grid cells per second, while a platform moving at even a modest speed crosses several. The sensor cannot clear ground as fast as the robot wishes to cross it, by a factor of several. Any architecture requiring every traversed cell to have been directly measured therefore yields a robot that crawls or stutters. The design accommodates this rather than attempting to defeat it.

3.4 Pose Resolution: Giving the Detector Its Own Frame

The detector is mounted low and well forward of the chassis centre — by a fixed offset of [redacted]. This placement is deliberate: detection efficiency falls off with distance from the surface, so the detector wants to be near the floor, and ahead of the chassis so the platform can measure ground *before* driving onto it. But it creates a registration problem that is easy to get wrong and hard to diagnose. If each reading is recorded at the robot's pose — the obvious first implementation — every reading is misplaced. Worse, the error is not a constant offset but a *rotating* one: as the platform turns, the detector sweeps an arc around the chassis centre, so the direction of the error changes continuously with heading. The resulting map is smeared in a way that looks convincingly like sensor noise but is nothing of the kind — a map degraded this way does not announce itself; it is simply, quietly, plausibly wrong.

The correct treatment is to give the detector its own coordinate frame in the transform tree and let the transform library do the geometry, resolving each reading through a time-indexed lookup to where the detector actually was at the instant of measurement. Because the mapper asks the transform tree the question rather than hard-coding the geometry, the offset can later be made to *move* — a fact that becomes important for the future work of Section 5.3.

3.5 The Dual-Grid Mapper

The mapper resolves a genuine conflict: a scientific record and a path planner make incompatible demands of the same measurements. A record for human interpretation must be conservative — it must never claim to have measured ground it did not, so that an operator can see at a glance what carries evidence and what does not. A planner input must be dense enough that the robot can actually move, which means filling in some ground the detector did not directly sample. A single grid cannot be both without being either untrustworthy or unusable.

The resolution is to refuse the compromise and maintain *both*: a conservative map that records only ground the detector physically passed over, and a strict cost grid governed by a tiered authority scheme in which a direct measurement is absolute and permanent, while any inference used to keep the robot moving may fill only cells that are still unknown — never overriding a measured value. *The mapper's exact update rule, thresholds, and parameters are redacted (original Appendix, withheld).*

3.6 Hazard-Aware Navigation, and the Rest of the Stack

Because the planner's costmap is layered, the hazard field enters as one more cost layer: a cell measured as hazardous is made as costly as an obstacle, and unmeasured ground is declined. On top, 2D SLAM produces the occupancy grid, 3D reconstruction produces a dense reviewable model, and frontier exploration drives the survey to completion with minimal operator input. A finding during integration — discussed in the next chapter — forced the 2D and 3D mapping to run as separate sessions whose outputs are later fused under a region-ownership rule; that pipeline is described in outcome only, its internals redacted.

4 Results and Discussion

All quantitative results in this chapter are redacted. The original reports measured values, counts, durations, and field characterisations; here the numbers are withheld and only the qualitative conclusions and transferable lessons are retained. Black bars mark removed figures.

The validation was staged and bottom-up: each subsystem was checked against a specific, falsifiable criterion *before* the layer above it was trusted. This discipline matters because, in a system of this shape, a fault in a lower layer does not announce itself as a fault in that layer — odometry that lies quietly at the bottom of the stack presents, several layers up, as a hazard-mapping bug, and is easily debugged there, at length, in the wrong subsystem.

4.1 Driver Validation and Stability

The driver was validated for correctness, rate, and endurance. **Correctness** was established by direct comparison against the instrument's own display — an independent readout of the same measurement, and therefore an unimpeachable reference. This is what confirmed the decoding: no amount of internal consistency could have established it, since a systematically mis-scaled decoder is perfectly self-consistent. **Rate** was measured at approximately [redacted], consistent with the instrument's own update rate and confirming the driver adds no material latency. **Endurance** mattered most, because the failure mode being guarded against is intermittent and would pass any short test; the driver held a stable link across extended sessions — the longest continuous single-cell record ran to [redacted] consecutive readings, roughly [redacted] of uninterrupted acquisition — where an earlier, naive driver would have reported several phantom disconnections.

4.2 Field Characterisation and Source Response

The ambient field at the test location was characterised from [redacted] logged readings across [redacted] sessions, establishing a background of [redacted]. *The measured values, the session tables, and a notable calibration anomaly the append-only history caught are all redacted.* The more interesting qualitative result survives the redaction: a stationary detector measuring an unchanging field recorded a spread of roughly ±[redacted] about its own mean over several minutes — nothing in the environment changed; the *sampling* of it did, exactly as the Poisson character of the physics predicts. This is the empirical justification for two design decisions taken earlier on theoretical grounds: that a single reading cannot classify a cell, and that the per-cell record must retain more than a running mean.

Detector response was then verified against live sources of each principal radiation type under supervision, with the driver's published value tracking the instrument's display across elevated rates and the instrument's automatic ranging — confirming the sensing chain holds well beyond the narrow band of ambient background. One qualitative observation shaped the future work: the penetrating types were readily obtained at the mounted standoff, while the short-range type was acutely sensitive to the gap between source and detector window — as the physics predicts. This is a statement about the *robot's geometry*, not the instrument. *The trials were qualitative* *response checks, not a quantitative efficiency characterisation; that remains recommended before any output is used for a dose assessment.*

4.3 Pose-Resolved Mapping

The acceptance test for the transform chain was specific and falsifiable: *when the platform passes a source, the elevated cell must appear where the detector was, not where the chassis centre was.* The two differ by enough grid cells that the result is unambiguous — far too large to attribute to noise. The test passed, and it is more powerful than it first appears, because it exercises the whole chain at once: the transform tree must be complete and correctly parameterised, composed correctly through rotation; the timestamps must be sound; and the mapper must perform a time-indexed lookup rather than a current-pose one. A single wrong link anywhere displaces the result, so passing validates all of it simultaneously.

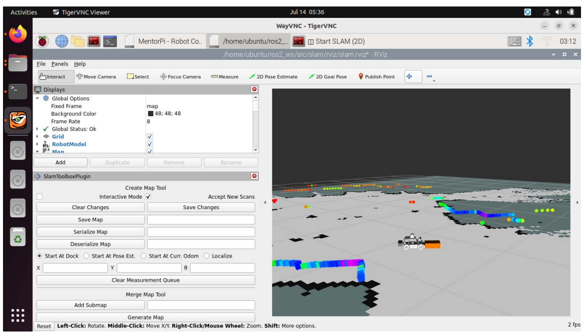
[Figure withheld — the live hazard-field map over the occupancy grid is redacted for confidentiality.]

[redacted] [redacted] [redacted] [redacted]

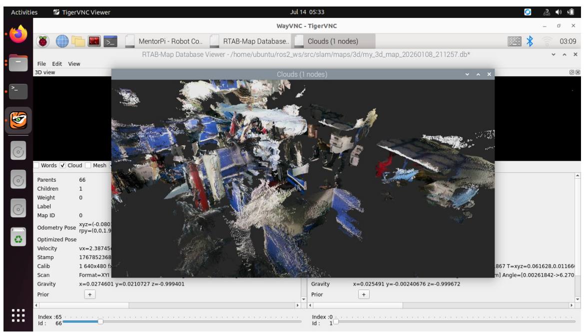
In the withheld figure, the measured field appears as a narrow *trail* tracking the path the platform drove, rather than a smooth field filling the room. That is not a deficiency; it is the map being honest. A conventional heat-map renderer would interpolate between measured points and produce a continuous, confident-looking field, most of which would be invention — a small detector sampling slowly physically cannot have measured it. The trail is a faithful statement of what was measured and, just as importantly, of what was not.

4.4 Mapping, Exploration, and a Negative Result

2D SLAM produced clean, metrically plausible occupancy grids with loop closure correcting drift on return to the start; maps were saved, reloaded, and reused. Autonomous frontier exploration ran unattended — identifying frontiers, dispatching them as goals, planning to them, and driving itself while extending the map. 3D reconstruction produced dense point clouds persisted to a reviewable database.



2D occupancy map built live.



Dense 3D reconstruction, reviewed offline.

Figure 4.1: General mapping outputs. Quantitative session details (keyframe counts, database sizes, map scales) are withheld.

The project's most instructive result was a negative one, and it is retained because the manner of its diagnosis is more valuable than the finding. Running the 2D and 3D mapping concurrently caused the 2D map to quietly stop updating — and *nothing else appeared wrong*: odometry stayed correct, no node crashed, nothing was logged. Every visible symptom pointed at the 2D mapper, and a considerable amount of time was spent there before the real cause emerged. Instrumenting the sensor topics under combined load revealed it: the LiDAR's publish interval was oscillating wildly, and the 2D mapper was receiving a fraction of the scan data it expected. The cause was CPU starvation — the 3D pipeline's appetite was preventing the LiDAR driver from servicing its device on schedule, and a serial device not read on time loses data irrecoverably.

The transferable lesson. A real-time task (servicing hardware on a schedule) and a throughput task (mapping, which can run late without loss) look identical to a default scheduler. Under load, the throughput task starves the real-time one, and the damage surfaces in a component two layers removed from the cause. Resource contention does not manifest where it originates; it manifests wherever the resulting data loss first becomes visible.

4.5 Hazard-Aware Navigation, and an Honest Caveat

Hazard-aware planning was implemented and demonstrated *during development*: with the hazard cost layer wired into the planner, the platform was observed treating a region marked hazardous as lethal and routing around it, exactly as it would a wall. This is the headline capability — it closes the chain from an undocumented serial port to a robot that changes its own behaviour based on what it has measured.

Stated precisely, and not overclaimed. The configuration in which this was demonstrated is *not* the one in the delivered snapshot. Late in the project the cost layer was unwired for an unrelated test and not restored before handover. In the delivered state the mapper publishes the hazard cost layer correctly and continuously — the producing side is complete and correct — but nothing consumes it, so the hazard-*navigation* mode drives without avoidance until the layer is rewired (a configuration task, not a research problem). The hazard-*mapping* mode is unaffected. A report that quietly implied otherwise would mislead precisely the reader most likely to rely on it.

4.6 Discussion

Taken together, the results establish that the complete chain is achievable on hardware of this class: an undocumented instrument was made to speak; its readings were given a position accurate to within a grid cell; those readings were accumulated into a map honest about what it does and does not know; that map was rendered into a form a planner can act on; and a planner was demonstrated acting on it — each link validated against a falsifiable criterion before the next was trusted. Three findings stand out as contributions beyond the working system: the general framing insight that converted an open-ended reverse-engineering problem into a configuration change; the dual-grid separation that refuses to force a scientific record and a planner input into one compromised grid; and the concurrency diagnosis, whose lesson — that contention surfaces far from its cause — is the project's most transferable.

5 Conclusions, Limitations and Future Work

5.1 Conclusions

The project set out to build a mobile robot that could map an unknown interior, measure the hazard field throughout it, register the two faithfully, and use the result to govern its own movement. All four capabilities were implemented and demonstrated on the physical platform. The broader conclusion is that the entire chain — from an undocumented serial port to autonomous, hazard-informed motion — is achievable on low-cost hardware of this class. That was not a foregone result: several intermediate steps, particularly the compatibility of a slow sensor with real-time planning, were genuinely uncertain at the outset.

5.2 Limitations

Stated candidly, as they bound what the results support:

1. **The hazard cost layer is unwired in the delivered configuration** — demonstrated in development, but not connected in the handed-over snapshot. This is the single most significant limitation and the first item of future work.
2. **Response verification was qualitative**, not a quantitative efficiency characterisation; survey output should not inform a dose assessment until that exists.
3. **The short-range radiation type is not usefully surveyed by the present geometry** — a geometric limitation of the robot, not the instrument.
4. **Trials were in a mock-up layout**; real interiors may present reflective metallic surfaces and lighting variation not yet tested against.
5. **A tethered link was descoped**, so the platform is not yet suited to the RF-restricted enclosures that partly motivated it.
6. **The holonomic chassis is under-exploited** — the controller does not yet use the base's ability to strafe, which for a survey robot is a real loss.
7. **Minor configuration defects remain**, documented in the internal report and [redacted] redacted here.

5.3 Future Work

The immediate task is to rewire the hazard cost layer into the planner and verify it against a synthetic hot zone before any test with a real source. Beyond that, the most substantive proposal is a **two-phase survey architecture** for the short-range radiation type. Because that type must be measured with the detector window nearly touching the floor — impossible while driving — the detector would ride retracted during a fast coarse pass that maps the penetrating types and finds candidate hot spots, then a fine pass would return to each candidate, stop, lower the detector, dwell, and take a proper measurement. This turns the machine from *a thing that draws a heat-map* into *a thing that finds contamination and then characterises it*. Crucially, the mapper needs no change: because it asks the transform tree where the detector was rather than assuming a fixed offset, making the offset move is handled automatically — the payoff of the pose-resolution design decision. Further architectural recommendations — protecting the real-time work from starvation, moving visualisation off the platform, exploiting the holonomic chassis, and hardening the power and compute — are set out in the internal report and summarised here only in outline.

Closing Note on What Is Withheld

If this report reads as light on specifics, that is by design. The detailed methodology, the instrument-protocol work, the measured data and results tables, the facility context and personnel, the system's reproduction appendices, and the in-situ imagery are all held confidentially and have been redacted from this release. What is presented here — the problem, the reasoning, the design decisions, and the transferable engineering lessons — is offered as an honest account of the *shape* of the work. I am glad to discuss the approach in general terms; I am not able to share the withheld material.

Bibliography

[1] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics.* MIT Press, 2005.

[2] H. Durrant-Whyte and T. Bailey. Simultaneous localisation and mapping: part I. *IEEE Robotics & Automation Magazine*, 13(2), 2006.

[3] T. Bailey and H. Durrant-Whyte. Simultaneous localisation and mapping: part II. *IEEE Robotics & Automation Magazine*, 13(3), 2006.

[4] P. J. Besl and N. D. McKay. A method for registration of 3-D shapes. *IEEE TPAMI*, 14(2), 1992.

[5] G. Grisetti et al. A tutorial on graph-based SLAM. *IEEE Intelligent Transportation Systems Magazine*, 2(4), 2010.

[6] W. Hess et al. Real-time loop closure in 2D LIDAR SLAM. *IEEE ICRA*, 2016.

[7] S. Macenski and I. Jambrecic. SLAM Toolbox. *Journal of Open Source Software*, 6(61), 2021.

[8] D. Scaramuzza and F. Fraundorfer. Visual odometry: part I. *IEEE Robotics & Automation Magazine*, 18(4), 2011.

[9] M. Labbé and F. Michaud. RTAB-Map. *Journal of Field Robotics*, 36(2), 2019.

[10] H. Taheri et al. Kinematic model of a four-mecanum-wheeled robot. *Int. J. of Computer Applications*, 113(3), 2015.

[11] R. Siegwart, I. Nourbakhsh, and D. Scaramuzza. *Introduction to Autonomous Mobile Robots.* MIT Press, 2011.

[12] S. Macenski et al. The Marathon 2: navigation system. *IEEE/RSJ IROS*, 2020.

[13] S. Macenski et al. Spatio-temporal voxel layer. *arXiv:2003.02308*, 2020.

[14] B. Yamauchi. A frontier-based approach for autonomous exploration. *IEEE CIRA*, 1997.

[15] G. F. Knoll. *Radiation Detection and Measurement.* Wiley, 4th ed., 2010.

[16] N. Tsoulfanidis and S. Landsberger. *Measurement and Detection of Radiation.* CRC Press, 4th ed., 2015.

[17] K. Nagatani et al. Emergency response to the Fukushima Daiichi accident using mobile robots. *Journal of Field Robotics*, 30(1), 2013.

[18] C. Ilas. Electronic sensing technologies for autonomous ground vehicles. *Int. Conf. on Electronics, Computers and AI*, 2013.

[19] M. Quigley et al. ROS: an open-source Robot Operating System. *ICRA Workshop on Open Source Software*, 2009.

[20] S. Macenski et al. Robot Operating System 2. *Science Robotics*, 7(66), 2022.

[21] T. Foote. tf: the transform library. *IEEE TePRA*, 2013.