King's College London

Development and Evaluation of a Robotic Radiation Monitoring System

Submitted by:

Naghim Ibragimov

K22031784

Project Report Submitted in Partial Fulfillment of the Requirements of the MSc Project Module

Engineering Department

Supervised by:

Prof. Kawal Rhode

London

August 2026

Declaration

I declare that parts of this submission have contributions from AI software and that it aligns with acceptable use as specified as part of the assignment brief and guidance and is consistent with good academic practice. The content can still be considered as my own words. I understand that as long as my use falls within the scope of appropriate use as defined in the assessment brief and guidance, then this declaration will not have any direct impact on the grades awarded.

I acknowledge use of software to:

- **Generate ideas or structure suggestions, for assistance with understanding core concepts, or other substantial foundational and preparatory activity.** Claude (Anthropic, `https://claude.ai`) was used to discuss the report's structure, to find and summarise published literature, to explain concepts encountered during the project, and to review my reasoning.
- **Write, rewrite, rephrase and/or paraphrase part of this essay.** Claude produced initial drafts of sections from material I supplied, which I then reviewed, corrected and rewrote.
- **Generate some other aspect of the submitted assessment.** Claude and Claude Code (Anthropic, `https://claude.com/claude-code`) were used for code assistance, review and debugging during development, and for offline analysis of the recorded data reported in Chapter 5. All experimental data were collected by me on physical hardware.

Abstract

Radiation surveys in hospitals are performed by hand. A physicist stands inside the controlled area holding a detector, moving between fixed positions and recording readings one at a time. The process is slow, produces only a few samples, and exposes the person responsible for radiation safety to the hazard being measured. A robot built previously at this institution already carries a detector through such rooms, but it only records: it cannot distinguish a place that was measured and found clean from one never visited, offers no principled way to choose where to measure next, and has no understanding of when the process is complete.

This project presents a four-legged robot that reasons about the radiation field rather than merely recording it. A statistical model estimates the radiation rate everywhere in a room together with its own uncertainty, using measurement noise that reflects the counting statistics of the detector carried. From that model the robot chooses each next measurement position by weighing the information expected against the distance it must walk, decides how long to remain there from the counts needed for a trustworthy reading, and ends the survey when no remaining measurement would repay the effort of reaching it. Every movement is proposed to an operator who approves, rejects or halts it. Because no sufficiently strong source was available during development, a simulated source was built that the rest of the system cannot distinguish from the real detector, allowing every experiment to be scored against an answer the estimating software cannot read.

The system was tested against real radioactive waste. In the final test it found the source to within twelve centimetres. That is about the size of the container holding it, and it matches the accuracy of published systems that use laboratory-grade equipment. The detector used here was uncalibrated and cost a few tens of pounds, and the source was weak, reading only about three times the natural background. An earlier test had three sources in the room but was scored against one of them. It found that source to within thirty-six centimetres. Analysis of the recorded data attributes much of the difference to the test conditions rather than to the method used. The most accurate result came from a test whose autonomous movement failed partway through, when the wireless link broke down. This is because the system draws its conclusions from the readings it has collected, not from finishing a planned route.

The work indicates that principled statistics, rather than expensive instrumentation, carries the accuracy of a robotic radiation survey, and that autonomy of this kind can be introduced in supervised stages that leave the operator in control throughout.

Table of Contents

List of Tables

Page

List of Figures

Page

List of Symbols

β Exploration weight in the acquisition function

ℓ Gaussian-process kernel length scale

λ Distance-penalty weight in the acquisition function

$\mu(\mathbf{x})$ Gaussian-process posterior mean (predicted dose rate)

N Accumulated detector counts at a measurement point

$\sigma(\mathbf{x})$ Gaussian-process posterior standard deviation (uncertainty)

ALARA As Low As Reasonably Achievable

CPM Counts Per Minute

DDS Data Distribution Service

EC Electron Capture

EI Expected Improvement

GM Geiger–Müller

GP Gaussian Process

ICP Iterative Closest Point

IDW Inverse Distance Weighting

IMU Inertial Measurement Unit

IRR17 Ionising Radiations Regulations 2017

MCL Monte Carlo Localization

NDT Normal Distributions Transform

PET Positron Emission Tomography

PI Probability of Improvement

ROS 2 Robot Operating System 2

SLAM Simultaneous Localization and Mapping

SNR Signal-to-Noise Ratio

TDD Test-Driven Development

TF Transform (ROS 2 coordinate-frame tree)

UCB Upper Confidence Bound

Chapter 1: Introduction

1.1 Background and Motivation

Radiation monitoring is a routine requirement in hospital X-ray and radiotherapy rooms. Under the Ionising Radiations Regulations 2017 (IRR17), employers must restrict occupational exposure, and surveys are governed by the ALARA (As Low As Reasonably Achievable) principle [1]. In current NHS practice these surveys are performed manually: a medical physicist stands at fixed positions inside the controlled area, holding a handheld dosimeter for one to two minutes per point. The process is repetitive and time-consuming, produces only a sparse set of point measurements rather than a full spatial picture of the room, and exposes the very people responsible for radiation safety to the hazard they are measuring.

Mobile robots offer a direct answer to this issue: if a robot carries the dosimeter, the physicist can supervise from outside the irradiated zone while the robot collects spatially referenced measurements with far denser coverage than manual spot checks. Robotic radiation surveying is well established in the nuclear industry, but the platforms used there are either wheeled vehicles unsuited to cluttered clinical rooms or legged platforms costing tens of thousands of pounds, and applications in healthcare remain largely unexplored; Chapter 2 reviews this landscape.

The long-term aim of the work at St Thomas's Hospital, of which this project is part, is a fully autonomous system: a robot placed in an unfamiliar radiation room that explores it by itself and concentrates its measurement effort where the radiation actually is, returning to hot or uncertain areas of its own accord rather than following points that were picked by a human.

1.2 The Prior System and Its Limitations

This project builds on an existing radiation-monitoring system developed at King's College London around the Unitree Go2 quadruped, a platform costing roughly one-eighth of the Boston Dynamics Spot used in comparable industrial work [2, 3]. The prior system carries a J305 Geiger–Müller sensor and was validated in a controlled X-ray room at St Thomas' Hospital, where its readings achieved a Spearman rank correlation of $\rho = 0.96$ against a professional scintillation dosimeter across 14 waypoints [2].

Its Teach-and-Repeat workflow, however, imposes three separate manual stages on every survey: drive the robot around to build a LiDAR floor plan, walk it to each measurement position to record a waypoint route, then replay the route autonomously, stopping at each point to measure, with heatmaps generated offline afterwards. This design has structural consequences. Mapping and recording must occur within a single boot session, because the odometry frame resets on power-up, cross-session reuse of a map was observed to fail with offsets of 0.5–0.7 m [2]. Radiation is measured only at specific waypoints that were selected, so the space between them is never sampled. And although the operator dashboard displays live readings, the spatial radiation picture is never constructed live.

1.3 From Recording to Reasoning: The Problem This Project Addresses

Beneath these workflow limitations lies a deeper one. The prior system, and indeed most fielded radiation robots, *records* radiation: it stores what was measured where, and interpolates between

those points for display. It cannot distinguish a location that was measured and found cold from one that was never measured at all; it offers no principled way to choose where to measure next, leaving that to human operator; and it has no notion of completion, so it ends when the operator decides to stop rather than when the data is sufficient.

This project addresses that gap. Its central claim is that the robot should not merely record a radiation field but *reason* about it. It should build a statistical model of the invisible field and know how uncertain that model is at each point. It should judge every candidate measurement by the information it would gain against the effort of reaching it. It should measure for exactly as long as the counting statistics require, and stop when it can show that nothing worth measuring remains. Throughout, a human should retain authority over every movement.

Achieving this requires two foundations the prior system lacked: live access to the robot's sensors, and localization that can be trusted across a survey.

1.4 Contributions

The contributions of this report are as follows.

A passive live-survey architecture. After establishing with evidence that neither direct DDS nor WebRTC access to the robot is available from an external machine, a ROS 2 was used that consumes the robot's existing Jetson TCP bridges strictly read-only and republishes pose and LiDAR into ROS 2, collapsing the prior three-stage workflow into a single live run. Because it possesses no command channel, it cannot conflict with manual control, demonstrated on hardware with point clouds accumulating to the order of 10^5 points while handheld driving continued unaffected.

A corrector-agnostic live mapping system. The dense floor plan is accumulated in a world frame selected at launch, so raw leg odometry, 2D pose-graph SLAM, 3D LiDAR odometry or map-based relocalisation can be substituted without changing the mapping code. This decoupling makes drift comparison a controlled experiment, and became the essential integration that was needed for the whole system.

A localization-drift investigation. Leg odometry, slam_toolbox and KISS-ICP were evaluated on this platform; a configuration defect that silently sabotages KISS-ICP indoors was identified and corrected; and the analysis motivates a two-phase architecture in which a reference map is built once and the robot then localises against it.

Live radiation mapping fused with the floor plan. Geiger readings are paired with the robot's pose, so radiation and map share coordinates by construction, and exported as a survey log, a live heatmap and a combined overlay. A calibration discrepancy documented but unresolved in the prior work was traced and provisionally corrected.

The Smart Radiation Scan: an intelligence layer, designed, implemented and validated. A Poisson-aware Gaussian process models the field and its uncertainty; a distance-penalised acquisition function proposes each next measurement; dwell time is derived from counting statistics rather than guessed; a principled stopping rule ends the survey on statistical evidence; and the robot walks to its own chosen points under a (hold-to-walk) human controlled obstacle freezing. No source of sufficient strength was available for development, so a simulated one was built. It computes readings

from the robot's real position using the physics of the source field, and delivers them in exactly the format the real detector uses, so nothing downstream can tell the difference. The true position is held in a file the estimating code cannot read. Every run therefore has a known answer, and its error can be measured in metres.

A verification record. The intelligence layer was built test first, across fourteen tasks that were each rreviewed independently. The test suite grew from 72 tests when the build was finished to 94 by the end of it. Review before deployment found three safety-critical defects. A further eleven faults appeared during field operation and were corrected, each secured by a regression test where one could be written.

1.5 Scope and Attribution

This report describes the author's own work, contained in the `automation/` codebase. It builds on inherited infrastructure from the prior system, the Jetson-side DDS-to-TCP bridges and their client libraries, and the occupancy-grid logic, which was refactored rather than written anew. The project sits within a wider team effort in which other members are responsible for map-based relocalisation, design cad parts and the operator dashboard; those components are described only where the system's operation requires it, and are attributed explicitly wherever they appear.

1.6 Report Outline

Chapter 2 reviews radiation monitoring in healthcare, robotic surveying, localization theory, and the informative-sampling and supervised-autonomy literature on which the intelligence layer rests. Chapter 3 formulates the problem, shows why simpler approaches fail, and defines the research questions, assumptions and objectives. Chapter 4 presents the methodology: the live survey architecture, the drift-correction evaluation, the field model, the acquisition, dwell and stopping rules, the supervised execution modes, the safety architecture, the validation methodology and the implementation and verification approach. Chapter 5 reports and analyses the results, evaluating the work against its objectives. Chapter 6 concludes with the impact of the work and directions for further development.

Chapter 2: Literature Review

2.1 Radiation Monitoring in Healthcare

Ionising radiation in hospitals is regulated in the United Kingdom by IRR17, which obliges employers to restrict occupational exposure and to designate and monitor controlled areas [1]. Operationally, compliance rests on the ALARA principle. Routines are conducted manually by medical physicists using handheld dosimeters at fixed grid positions, slow, spatially sparse, and self-contradictory in that the person ensuring radiation safety absorbs dose to do so. In nuclear medicine the hazard extends further: patients administered radiopharmaceuticals are themselves mobile sources, and contamination can spread to corridors and waiting areas [4, 5]. Recurring surveys, structured indoor environments and a strong incentive to remove humans from the measurement loop make this a natural target for automation, yet, as the next section shows, it remains among the least developed application areas.

2.2 Robotic Radiation Surveying: State of the Art

Robotic radiation surveying is most mature in the nuclear industry. CARMA II surveys alpha, beta and gamma radiation fully autonomously, using LiDAR-based SLAM, specifically slam_toolbox, for mapping and localization, and integrating radiation awareness into its navigation costmaps [6]. Sattar et al. demonstrated a low-cost wireless robot for radiation detection, but it is teleoperated and its readings are not spatially referenced, the operator learns *that* radiation is present, not *where* [7]. Gabrlík et al. combined ground and aerial vehicles for outdoor surveys, reporting source-localization errors of 0.10 m for the ground platform against 2.82 m from the air; critically, their localization relies on RTK-GNSS, unavailable indoors [8], so indoor systems must localise from their own sensors, exactly where the drift issue appears in this project.

Legged platforms extend research into cluttered, uneven environments that defeat wheeled bases. Lawrence Berkeley National Laboratory mounted a gamma-ray imaging system on a Boston Dynamics Spot [9], and a Unitree Go1 has performed radiation inspection in the LHC tunnels at CERN [10]; both target industrial facilities, and the Spot platform's cost (approximately £60,000) is incompatible with routine hospital deployment. Healthcare-specific efforts remain embryonic: RadRob15 is a wheeled platform that alarms on contamination but neither maps nor localises it [4], and Winiarski et al. survey assistive-robot concepts for nuclear medicine in which radiation monitoring is identified but not yet realised with a mounted sensor [5]. Table 1 summarises these systems.

2.3 The Prior KCL System

The immediate foundation for this project is the KCL Go2 radiation-monitoring system [2, 3]: a Unitree Go2 EDU (L1 LiDAR, IMU-based leg odometry, front camera) carrying a J305 Geiger–Müller tube read by an Arduino, with an onboard Jetson bridging the robot's internal network to the operator's WiFi. Its Teach-and-Repeat workflow uses a waypoint controller and a motorised gantry for dual-altitude measurement, with heatmaps generated after the run by inverse distance weighting [14]. Validation at St Thomas' Hospital detected the source location (peak 13.9 μSv/h) and achieved $\rho = 0.96$ against a co-mounted professional dosimeter [2]. The design deliberately avoided real-time SLAM to reduce

Table 1 Robotic radiation surveying systems compared with the prior KCL system and this work. “Ref.” indicates whether readings are spatially referenced, “Live” whether the map is built during the survey, and “Adaptive” whether the system chooses its own measurement locations.

System	Platform	Environment	Localization	Ref.	Live	Adaptive
CARMA II [6]	Wheeled	Industrial	SLAM	Yes	Yes	No
Sattar et al. [7]	Wheeled	Laboratory	None	No	No	No
Gabrlík et al. [8]	UGV+UAV	Outdoor	RTK-GNSS	Yes	No	No
LBNL Spot [9]	Legged	Industrial	SLAM	Yes	Yes	No
RadRob15 [4]	Wheeled	Nuclear med.	None	No	No	No
West et al. [11]	Wheeled	Reactor	SLAM	Yes	—	Post hoc
Adams et al. [12]	Legged	Laboratory	SLAM	Yes	Yes	Yes
Lazna & Zalud [13]	Simulated	Outdoor	RTK-GNSS	Yes	—	Yes
Prior KCL [2]	Legged	Hospital X-ray	Leg odometry	Yes	No	No
This work	Legged	Radiation test	Evaluated	Yes	Yes	Yes

computational load, accepting the constraints described in Section 1.2, which define this project’s starting point.

2.4 Localization, Mapping and Drift

A robot estimating its pose purely by integrating own motion is performing dead reckoning, and it grows without limit. Each step’s small error is integrated into all subsequent poses [15]. Legged odometry is a refined form of this, fusing joint kinematics and inertial data; it is smooth and locally accurate, but foot slip and IMU bias make global drift unavoidable [16, 17]. The consequence for mapping is characteristic: the map looks locally correct everywhere, but revisited structure is redrawn slightly displaced, ghosted walls, doubled corners.

Simultaneous localization and mapping addresses this by estimating the map and the position together. Its key mechanism is loop closure: recognising a place that has already been mapped, and correcting the error that has built up since [15, 18]. Bailey and Durrant-Whyte separate the classical state-space formulation from trajectory-based approaches, in which positions form a graph and sensor scans are aligned to build the map [18]. That distinction classifies the tools evaluated here. slam_toolbox is graph-based two-dimensional SLAM with loop closure [19]. KISS-ICP aligns each scan to those before it but never closes a loop [20], so it reduces drift without being able to bound it.

A third approach avoids drift altogether by matching live sensor data against a map that already exists. Monte Carlo localization represents the robot’s belief about its position as a set of particles [21], and NDT-MCL applies this to a map built from local Gaussian distributions, reaching accuracy comparable with fixed positioning infrastructure [22]. The three families differ in one important respect: odometry drifts, SLAM corrects itself at loop closures, and map-based localization does not drift at all. That difference shapes both the evaluation in Chapter 5 and the two-phase approach adopted here.

2.5 Field Estimation and Informative Sampling

Estimating a radiation field from sparse, noisy measurements is a regression problem with a specific measurement model. Ristic et al. established the canonical treatment for low-cost Geiger–Müller counting: Poisson-distributed counts over a background plus inverse-square source superposition, with Bayesian estimation of source number, position and intensity [23]. Gaussian process regres-

sion is the established method for mapping the resulting field, validated for robot-collected radiation mapping of a nuclear reactor, and, crucially, returns not only a predicted field but its uncertainty [11].

That uncertainty is what makes measurement *planning* possible. In informative path planning, the next measurement is chosen to maximise expected information, typically through an acquisition function balancing predicted value against predicted uncertainty; frameworks in environmental monitoring establish the replan-after-each-measurement loop and the practice of reporting field error against distance travelled [24, 25]. Applied to radiation, Adams et al. demonstrated Gaussian-process adaptive sampling with a probability-of-improvement acquisition on a quadruped, converging on a caesium source within roughly fifteen samples, and showed that dynamically inflating the exploration parameter after unreachable points prevents failure loops [12, 26]. Miller et al. argue that the robot's walking cost belongs inside the acquisition itself, proposing a distance-penalised upper-confidence-bound with a heteroscedastic surrogate reflecting that Poisson noise grows with intensity [27].

Two cautions come from the literature. First, purely information-greedy behaviour misses weak sources: Lazna and Zalud's particle-filter survey found strong sources in essentially every run but the weakest in only 62.8%, against near-certainty for a systematic sweep, while still reducing measurements by 39% and time by 41–44% [13]. Coverage pressure must therefore be retained alongside exploitation [28]. Second, measurement duration matters: Gabrlík et al.'s speed-versus-sensitivity analysis makes explicit that integration time governs statistical quality [8], and for Poisson counting the relative standard error of N counts is $1/\sqrt{N}$ [29], yet published robotic systems use fixed dwell times.

Finally, a survey has to end. Most published adaptive surveys stop when a fixed budget of time or distance runs out, and show afterwards that the estimate had already converged [30]. Better criteria decide this during the survey itself. One approach stops when the information still available falls below a threshold [30]. Another watches the expected improvement from further measurement and stops once it stagnates [31].

2.6 Supervised Autonomy

Full autonomy is not automatically desirable in safety critical environment. Chiou et al. formalise variable autonomy, distinguishing operator-initiated from mixed-initiative control switching, and demonstrate quantitatively that shared control can outperform both pure teleoperation and full autonomy [32]. Their field exercise with CBRN personnel conducting robot-assisted radiation survey found that operators require predictable behaviour, the ability to interrupt, and transparency about what the robot intends to do next [33]. Surveys of variable autonomy confirm that such performance and workload Surveys of variable autonomy confirm these performance and workload benefits [34]. [34]. This evidence reframes human gating not as an admission of incomplete autonomy but as a deployment requirement, and it motivates the interaction design adopted in Section 4.8.

2.7 Validation Without a Strong Source

Developing an adaptive survey algorithm requires a radiation field strong enough to plan against, which the weak sources available in a student laboratory cannot provide. The common solution is to simulate the field, at a level of realism. Wright et al. showed that analytic models of radiation and its absorption, checked against full particle-transport simulation, are accurate enough for research on

robotic inspection [35]. Proctor et al. went further and described scenarios by their signal-to-noise ratio rather than by source activity, which gives a principled way to make a test harder or easier [36]. Full particle-transport simulation is reserved for final validation, as in the physics-informed localization work of Son et al. [37, 38].

2.8 Research Gaps Addressed

Three gaps emerge from this review, and this work addresses each of them.

First, no published study compares the available acquisition functions on an indoor radiation task constrained by an occupancy map. Existing systems choose one, such as probability of improvement or upper confidence bound, without comparing it against the alternatives [12, 27].

Second, no published system asks the operator to approve each individual waypoint. The closest work either switches between levels of autonomy [32] or chooses measurement points entirely on its own [12].

Third, no paper addresses how long a Geiger–Müller tube on a robot should measure at each point. The literature either fixes the duration in advance [26] or discusses the difference between speed and sensitivity as a whole [8].

A fourth contribution follows from how the system was validated. Existing studies that use simulated radiation sources run entirely in simulation [35, 36]. Here only the radiation is simulated: the robot, the room and the position estimate are all real, so every run can be scored against a known answer and its error measured in metres.

Chapter 3: Problem Formulation

3.1 Problem Definition

The problem can be stated as follows. The robot is given a floor plan, a continuous estimate of its own position, and a limited amount of time. It takes readings $z_i = (x_i, y_i, c_i)$, each recording a count at a chosen position, where the counts follow Poisson statistics. From these it must do four things: estimate the radiation field $f(x, y)$ across the room together with its own uncertainty at every point; identify where the strongest source lies; choose each next measurement so that the information gained is worth the effort of reaching it; and stop once no remaining measurement is worth taking. Two constraints apply throughout. Measurements may only be taken at positions in free space that can be reached by a safe route, and every movement must be approved by a human operator.

This is a problem of informative path planning, sometimes called adaptive sampling. What distinguishes this instance of the problem is is the physics of a radiation field, the statistics of a detector that records very few counts, and the requirement that a person supervises each move the robot makes.

3.2 Why Simpler Approaches Fail

Each design decision in this system answers a documented failure of a more obvious approach. Table 2 sets these out. Taken together, they show that the difficulty lies not in moving the robot, but in deciding where it should measure.

Table 2 Simpler alternatives and the documented reasons they fail.

Simple approach	Why it fails	This work instead
Exhaustive grid sweep	At room scale, of order 600 dwell points, some 2.5–4 h, beyond the platform's battery endurance, a limit encountered directly during testing; yields no uncertainty statement and no early answer [28]	Spend a finite budget where information gain is highest
Drive toward the highest reading	Greedy source-seeking is trapped by local maxima and misses secondary sources: 62.8% detection of the weakest source [13]	Uncertainty term plus explore-inflation retains coverage pressure
Interpolate readings (IDW)	Cannot distinguish "measured cold" from "never measured"; no principled next point; no stopping notion [2]	Gaussian-process posterior mean *and* variance
Fixed dwell at every point	Wastes time where the field is strong, yields untrustworthy statistics where it is weak; published systems hard-code a duration [26]	Dwell derived per point from counting statistics
Stop when the operator tires	Unquantifiable, unrepeatable, indefensible	Stop on an expected-improvement floor [31]

3.3 Research Questions

RQ1. Can floor-plan construction and radiation mapping be performed live, in a single manually driven run, using only passive access to the robot's sensor data and without interfering with manual control?

RQ2. How does leg-odometry drift manifest at survey scale, and can available correction methods correct it without degrading the map quality that clean leg odometry already provides?

RQ3. Can a quadruped robot plan its own radiation measurements, estimating the field with calibrated uncertainty, selecting points of interest under a movement cost, dwelling according to counting statistics, and terminating on statistical evidence, under human supervision, and how accurately does it localise a hotspot?

3.4 Assumptions

The work rests on several assumptions. The environment is indoors, structured and unchanging during a survey. A wireless network links the robot's onboard computer to the operator's computer. The detector is a fixed piece of hardware, a Geiger–Müller tube reporting once per second, so the contribution of this project lies in software and method rather than in instrumentation. The bridge running on the robot's onboard computer is the only route by which sensor data can reach the operator, direct connections having been shown not to work. Finally, for validating the intelligence layer, a simulated field following the inverse-square law with Poisson noise is assumed to be an adequate substitute for a real source at this stage of research [35, 36].

3.5 Objectives

The objectives of this work are:

- O1 Read position, LiDAR and radiation data live from the robot without interfering with manual control.
- O2 Build a floor plan that updates continuously during a survey, with the source of position information selectable at launch.
- O3 Pair each radiation reading with a position as it is taken, and align the resulting outputs to the floor plan.
- O4 Measure how far leg odometry drifts, evaluate methods of correcting it, and recommend an architecture.
- O5 Design and build the smart radiation scan: a field estimate with uncertainty, a way of choosing measurements that accounts for their cost, a measurement duration derived from counting statistics, and a principled rule for stopping.
- O6 Carry out the chosen measurements autonomously under human supervision, with layered safety.
- O7 Validate the intelligence layer against known source positions, reporting the error in metres.

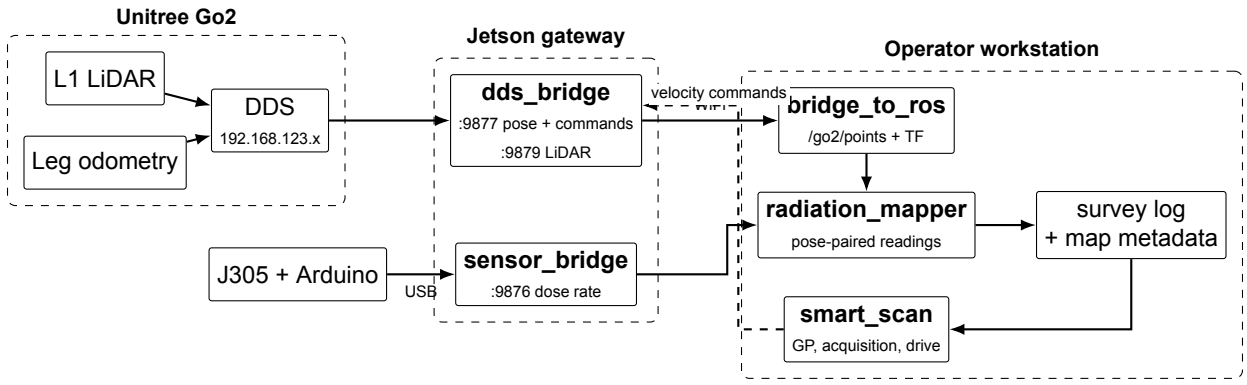

Figure 1 System architecture. Sensor data leaves the robot's isolated internal network only through relay services on the onboard gateway, which the workstation consumes over plain TCP. In manual surveys the connection is read-only; in supervised autonomous operation the dashed return path carries velocity commands under the safety protocol of Section 4.9.

Chapter 4: Methodology

4.1 System Overview and the Three-Walk Pipeline

The system spans four machines, shown in Figure 1. The Unitree Go2 EDU publishes LiDAR data cloud and leg-odometry state over DDS on its internal wired network (192.168.123.x), physically unreachable from any external device. The onboard Jetson takes this network and the external WiFi network, running the prior system's bridge services: LiDAR relay (TCP 9879, $\sim$5 Hz), pose and authenticated command channel (9877, $\sim$10 Hz, single client), and Geiger readings (9876, 1 Hz). A workstation VM (Ubuntu 22.04, ROS 2 Humble) runs this project's pipeline. Where system is driven manually, a phone running the Unitree application is the sole commander.

Operationally the system is organised as three walks. *Walk 1* produces the floor plan. *Walk 2* is a coarse radiation pass, driven manually, which pairs readings with position and logs them. *Walk 3*, the Smart Radiation Scan, loads that coarse log and refines the survey adaptively: the robot proposes each next measurement, the operator approves it, the robot measures and updates its model, until the survey terminates. This coarse-then-targeted structure follows the only fielded precedent for two-stage radiation surveying [39], with the difference that revisit locations are chosen by a principled acquisition function rather than by thresholding.

4.2 Passive Listening over the Jetson Bridge

Sensor data reaches the computer through a relay that the prior system already provided: because the robot publishes over DDS on an internal net that no external device can reach, a service on the onboard Jetson subscribes to those topics and re-transmits them over plain TCP. This arrangement was inherited and functional at the outset of the project.

An alternative was also evaluated. The community `go2_ros2_sdk` driver provides a ROS 2 integration written specifically for this robot, together with slam_toolbox and a navigation stack. Had it worked, it would have supplied drift-corrected mapping and localization ready-made, saving considerable effort. It did not work. Both of its connection modes failed on this platform. The first could not reach the robot's internal network from the workstation, confirmed by inspecting the network interfaces and by

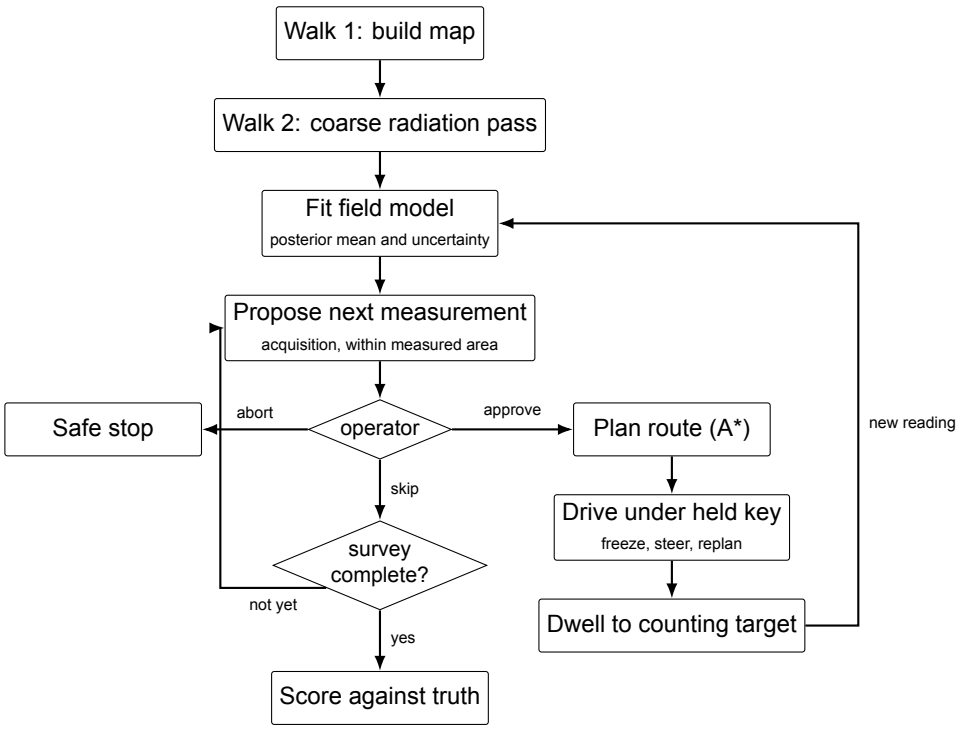

Figure 2 The supervised survey loop. Every measurement passes through the operator gate before the robot moves, and each accepted reading updates the field model, which in turn determines the next proposal. The survey terminates on its own statistical criterion rather than on a fixed budget.

pings that never returned. The second was refused by the robot, which rejected the connection on the required port, and a port scan showed all its ports closed. That mode would in any case have required the phone application to be shut down, which conflicts with driving the robot by hand. The decision was therefore to keep the working bridge and build the ROS 2 integration on top. This was a negative result, but a documented one, and it determined the architecture of everything that followed.

A second constraint shaped the design just as strongly. During a manually driven survey the handheld controller must be the only thing commanding the robot, so no part of the survey system may compete with it for control.

The author's contribution begins on the workstation. `bridge_to_ros.py` connects to the two sockets on separate threads. One receives the LiDAR data and reconstructs each scan into an array of points, discarding invalid and near-zero returns. The other keeps track of the most recent position. The ROS 2 node then minimizes the number of points, publishes them for other parts to use, broadcasts the robot's position ten times a second, and accumulates points into the map five times a second. In manual mode both connections carry data in one direction only. The workstation sends nothing back, so the listener cannot interfere with manual control even in principle.

4.3 Live Floor-Plan Reconstruction

The floor plan is rebuilt from scratch on a rolling cycle rather than added bit by bit. The node keeps a growing set of points in world coordinates. Each new scan is transformed into the chosen coordinate frame, added to the set, it's then thinned by rounding points onto a 5 cm grid and removing duplicates, which keeps memory use bounded. Roughly every two seconds the whole set is passed to the grid builder, a self-contained function extracted from the earlier system's mapper and covered by unit tests.

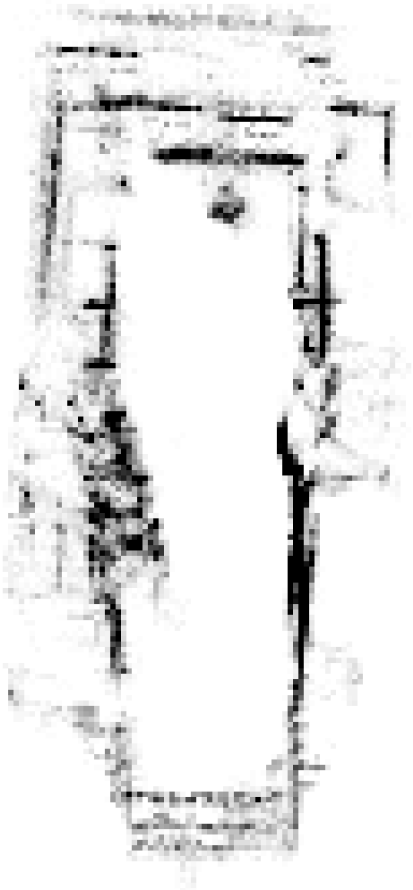

Figure 3 A floor plan produced by the live pipeline once localization was sufficient, shown at the 0.05 m cell resolution used throughout. Scattered returns along the left edge are furniture and clutter within the room rather than positional error. Compare with Figure 4, which shows the same process running on an uncorrected pose estimate.

The builder keeps only points between 0.1 m and 1.5 m above the floor, discarding the floor and ceiling so that only walls and furniture are shown. Choosing this band correctly proved important: too wide a range lets floor and ceiling returns smear across the image. The remaining points are flattened onto the horizontal plane and sorted into cells 0.05 m across, with each cell counting how many points fall inside it. A safety limit coarsens the resolution if a corrupt reading would otherwise produce an enormous grid.

The counts are then drawn (black-white). Any cell with eight or more points is rendered fully black. A fixed scale was chosen over the percentile-based one used by the offline mapper because it keeps successive rebuilds looking stable rather than shifting in brightness as data accumulates. Alongside the image, the builder records the grid's origin, resolution and coordinate frame. This is what allows the radiation heatmap, and any future display, to line up with the floor plan exactly.

Two design choices deserve justification. Rebuilding the whole map each time, rather than adding to it, costs a little computation but keeps everything up to date (easy to verify). A corrupted scan cannot permanently damage the map, and switching to a different source of position takes effect at the next rebuild. Counting points per cell also filters noise naturally: a wall is hit by many scans, while a stray

reflection is hit once.

The most important property is that the builder does not care which source of position it is given. The coordinate frame is chosen at launch, so the same code produces the map from raw leg odometry, from slam_toolbox, or from KISS-ICP. This makes the drift comparison a fair test, since the position source changes while the rendering never does, and it became the interface on which the rest of the system was built.

4.4 Live Radiation Mapping and Fusion

The radiation chain begins at the J305 Geiger–Müller tube, wired to an Arduino Mega. The firmware counts pulses using a hardware interrupt, keeps a rolling 60-second window of one-second totals, converts these to counts per minute using the true elapsed time, and reports one dose rate every second. The sensor bridge on the robot forwards these values to the computer.

The conversion factor needs an honest note. The manufacturer's figure of 153.8 counts per minute per microsievert per hour was found by the earlier project's own cross-check to read about 2.4 times too high against a professional dosimeter [2]. This project therefore uses a corrected value of 369.12, which closes that ambiguity. It remains a prototype estimate rather than a traceable calibration. At background levels the one-second counts are almost always zero, so individual readings are very coarse. This is the nature of counting statistics rather than a fault, and the firmware's rolling window smooths it over a longer real time base without inventing data.

`radiation_mapper.py` obtains the robot's position from the transform tree rather than by opening a second connection to the position port. There are two reasons. Practically, that port accepts only one client, so a second connection would disconnect the mapping bridge. Architecturally, using the transform tree means the mapper automatically uses whichever corrected frame the floor plan is built in, so the radiation data and the map share coordinates by construction rather than by calibration.

Once a second, the current smoothed reading is paired with the current position and added to the survey log. The growing set of readings is interpolated for live display using inverse distance weighting [14], masked so that unmeasured areas stay transparent, and drawn over the floor plan. The log itself, holding a timestamp, a position and a value for each point, is the primary output and the single source of truth used by the intelligence layer. All outputs are cleared when a run starts, so a survey can never display data from an earlier one.

Two limitations should be noted. A reading is paired with the most recent position rather than matched by timestamp, so a reading taken while the robot is moving is attributed to a single point and pushed across the ground it covered. And the map is a flat two-dimensional view, while dose rate also varies with height. The earlier system addressed this by raising the detector on a gantry at some waypoints [2].

4.5 Drift-Correction Methods Evaluated

The baseline is the map built from leg odometry. An early and important observation was that this map is clean. The walls are smooth and the room is recognisable. Its only fault is that the position estimate drifts slowly, so a corner revisited multiple times is drawn slightly out of place (Figure 4).

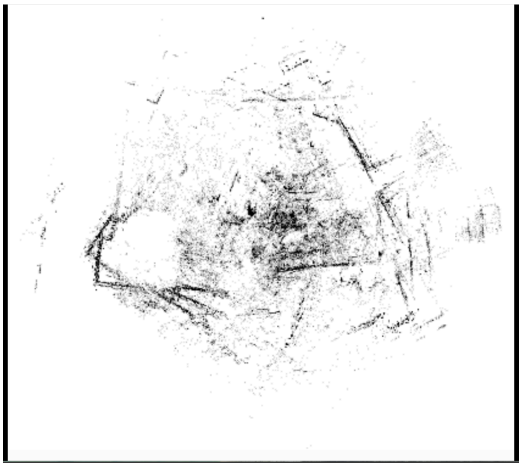

Figure 4 The drift problem made visible. A floor plan built in the leg-odometry frame: the geometry is correct locally, but structure the robot returns to same point many times, leaving doubled and smeared walls. The map is clean everywhere and wrong globally.

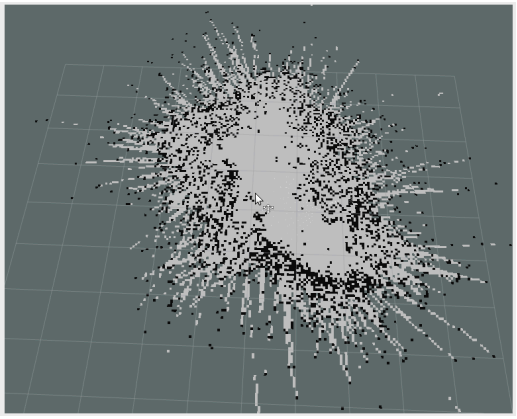

Figure 5 A failed two-dimensional mapping attempt. Flattening a sparse three-dimensional scan into a thin flat one left scan matching too weak to align successive scans consistently, so wall structure is smeared outwards instead of resolving into a room. Diagnosing failures of this kind, rather than the algorithms themselves, occupied much of the localization work.

Any correction that introduces jitter therefore makes the map worse than no correction at all. The requirement is not simply to reduce drift, but to reduce it without losing smoothness.

slam_toolbox. The 3D scan was flattened into a flat one and passed to slam_toolbox [19], with the map built in the corrected frame it produced. The result was poor, and Figure 5 shows why. With most of the vertical structure thrown away, scan matching had too little to work with, successive scans were aligned inconsistently, and the walls were pushed outwards instead of resolving into a room. Using too wide a height band made this worse by letting floor and ceiling merge into this scan.

KISS-ICP. KISS-ICP [20] uses the published scans directly and estimates position by aligning each new scan to the map built so far. It was compiled from source for the workstation. The first results were poor, and the cause turned out to be configuration rather than the algorithm. KISS-ICP sets its cell size to one hundredth of its maximum range, and that range defaults to 100 m. The result is cells one metre across, which suits a vehicle driving outdoors but is far too coarse for a small room, producing unstable tracking and a smeared map. A further complication is that its launch file ignores individual parameters and accepts only a configuration file. The configuration was therefore limited to the size of

a room: a maximum range of 15 m, a minimum of 0.3 m, and motion compensation disabled because the new scans are already compensated. Since the cell size follows from the maximum range, this reduced it to roughly 0.15 m. That the defaults of an off-the-shelf tool can quietly ruin its indoor performance is reported as a finding in Chapter 5.

NDT-MCL. Matching live scans against a map made earlier cannot accumulate drift, because the reference never moves [21, 22]. This approach was implemented by another member of the team, using the same interchangeable interface described above. It is mentioned here because it leads to the conclusion below.

The two-phase conclusion. The outcome of this evaluation was a decision about architecture rather than a choice of tool: separate mapping from localization. The map is built once, and the robot then locates itself within that fixed map while the radiation survey runs live. This avoids the drift of live mapping entirely while keeping the system live, and it is the arrangement the intelligence layer assumes.

4.6 Field Model

Each reading pairs a position with a dose rate. The dose rate is converted to $y_i = \log(z_i + \epsilon)$, and the underlying field is modelled as a Gaussian process

$$f \sim \mathcal{GP}\big(m,\, k(\mathbf{x}, \mathbf{x}')\big), \qquad k(\mathbf{x}, \mathbf{x}') = \sigma_f^2 \left(1 + \frac{\sqrt{3}\, r}{\ell}\right) e^{-\sqrt{3}\, r/\ell} + \sigma_b^2, \tag{1}$$

where r is the distance between two points, ℓ is the length scale over which readings inform one another, and the constant σ_b^2 absorbs the unknown background level.

Three of these choices follow established practice. Gaussian-process regression is the standard method for mapping radiation fields measured by robots [11]. Working in logarithms stops the model predicting a negative dose rate, which is physically impossible but which an unconstrained model will produce near zero. And each reading is given its own noise level rather than sharing one. A reading built from many counts is precise; one built from two or three counts is not. Counting statistics give a variance of roughly $1/N_i$ for a reading of N_i counts [23, 27], so the model automatically trusts weak background readings less than strong ones. This matters more here than in most published work, which uses detectors far more sensitive than a Geiger–Müller tube.

The model produces two things at every free cell of the map. The first is a mean, $\mu(\mathbf{x})$, which is the estimated dose rate there. Together these form the radiation map, and the highest point is the estimate of where the source lies. The second is a standard deviation, $\sigma(\mathbf{x})$, which is how confident the model is at that point. This second output is what makes planning possible, and it is exactly what simple interpolation cannot give and essential for the intial stages of intelligent robot.

4.7 Choosing Measurements, Timing Them, and Stopping

Choosing where to measure next. Every candidate position on the map is given a score:

$$\alpha_{\mathrm{UCB}}(\mathbf{x}) = \mu(\mathbf{x}) + \beta\, \sigma(\mathbf{x}) - \lambda\, d(\mathbf{x}), \tag{2}$$

The score has three parts. The first rewards places where the model expects a high reading. The second rewards places where the model is unsure, with β controlling how much. The third subtracts the cost of walking there, over a distance d. Including travel in the score reflects the fact that walking is part of what a measurement costs [27]. The robot then goes to whichever point scores highest. Two alternative scoring rules can be selected instead, which is what allows the comparison identified as a gap in Section 2.8. Figure 6 shows the rule in operation.

Without a restriction, this rule fails in a specific way. The model is most uncertain where nothing has been measured, which means the highest-scoring point is usually outside the area surveyed so far. The robot is then sent away from the region it was asked to cover. This happened during the real test. Proposals appeared beyond the ground the operator had driven, and the robot could not reach them. It walked to the edge of the space it could traverse and stopped, because the target lay in an area the map showed as empty but which was not actually reachable. There are two reasons for this. Mapped free space at the edge of a survey is optimistic, since anywhere the sensor never looked is recorded as empty. And the check for a clear path treats the robot as a single point rather than as a body with width. A target just past the true edge of the room therefore looks valid. Candidates are now restricted to cells inside the measured area that can be reached by a clear straight path with a safety margin. This is the same restriction that Adams et al. note is missing from their own system [12]. If a proposal is skipped, or turns out to be unreachable, the weight on uncertainty is increased. This prevents the robot proposing the same unreachable point repeatedly, which is a known failure of fixed settings in cluttered rooms [12, 26].

Deciding how long to measure. A radiation reading becomes more reliable the longer it is taken, because more counts accumulate. For N counts the relative error is $1/\sqrt{N}$. Achieving a chosen relative error e therefore requires

$$N \geq 1/e^2, \qquad t_{\mathrm{dwell}} = \mathrm{clamp}\left(\frac{N}{\hat{\lambda}(\mathbf{x})},\ t_{\min},\ t_{\max} \right), \tag{3}$$

where $\hat{\lambda}$ is the count rate the model predicts at that point. The default target of 10% needs 100 counts. Dividing that by the predicted rate gives the time required, which is then limited to between 5 and 60 seconds. The robot therefore measures for longer where the signal is weak and less where it is strong. The duration comes from the physics of the detector rather than being chosen by hand. This adapts the treatment of speed and sensitivity used in field surveying [8, 29] into a rule that the robotics literature does not state directly.

Deciding when to stop. Each proposal carries an estimate of how much it would improve the model. When the best available improvement stays below a threshold for several proposals in a row, the survey ends [31]. The threshold is set as a fraction of the range of values seen so far, so it adapts to the room. In plain terms, the robot stops when nothing left is worth walking to. This measure falls as the model becomes more confident, hence the alternative approach of waiting until uncertainty levels off [30] is satisfied by the same test. The operator can stop the survey at any point, and a fixed time limit remains as a final bound.

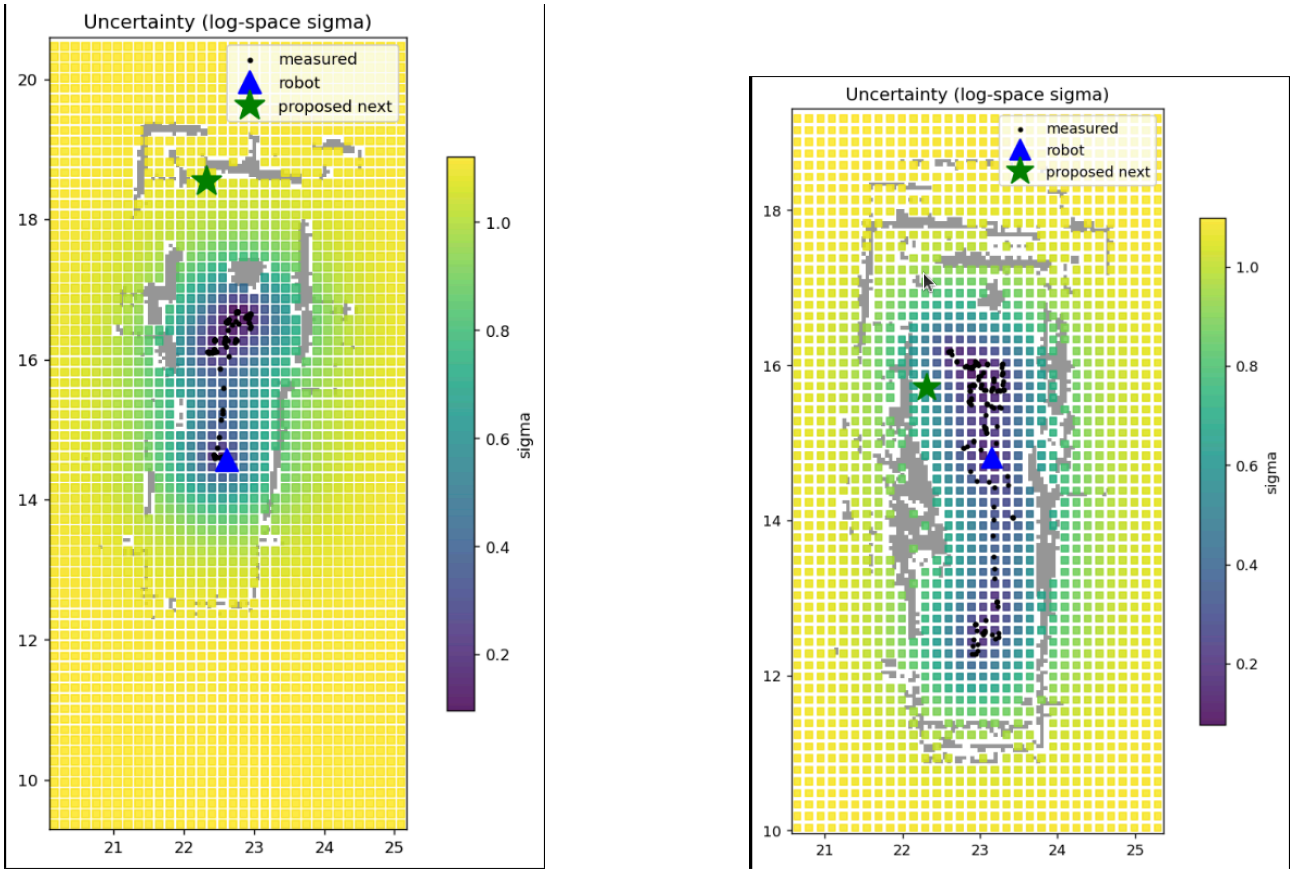

Figure 6 Why the restriction was necessary, shown on the two real-source surveys. The colours show the model's uncertainty: dark where readings exist, bright where the model is ignorant. Black dots are readings. Left: without the restriction, the proposal (star) is placed in unmeasured space well beyond the surveyed trail, and no route to it existed. Right: with candidates restricted to the measured area, the proposal sits at the edge of the data, where the model is uncertain but the map can be trusted.

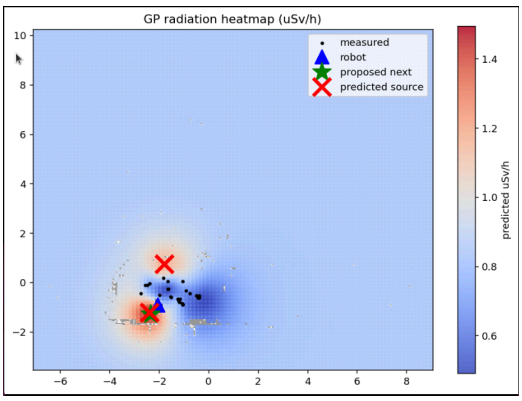

Figure 7 The operator's live display during a simulation rehearsal, showing the two marks the interface uses. Crosses mark positions where the model believes a source may lie, two appearing here. The star marks the next proposed measurement. The distinction matters in use: a proposal is a question the robot wants to ask, not an answer it is giving. Results obtained with real sources appear in Chapter 5.

4.8 Supervised Execution

At each step the system proposes one location and explains it in plain terms: the radiation reading it expects there, how uncertain it is, how far away it is, how long the measurement will take, and how close the survey is to finishing. It then waits. The operator can approve the proposal, skip it, or abort. Abort stops the robot and exits, at every stage of the process, and this is enforced by tests. Asking the operator to approve each individual measurement is the interaction identified as unpublished in Section 2.8. It follows directly from field studies showing that operators of robots in radiological work want behaviour that is predictable, interruptible, and clear about what the robot intends to do next [32, 33].

Figure 7 shows the display the operator sees. It uses two marks, and the difference between them matters. A star is the *next proposed measurement*: it shows where the robot wants to go, and makes no claim about radiation being there. A cross is a *predicted source position*: it shows where the model currently believes a source lies. In short, a star is a question and a cross is an answer. Figures recorded during the campaign show only stars, because source marking was added later.

The system runs in two modes, which share the same decision logic. In *suggest-only* mode the operator drives to each approved location by hand and confirms arrival. In *drive* mode the robot walks there itself. It plans a route using A* across the map, with obstacles expanded by the safety margin, and moves only while the operator holds a key. Releasing the key stops it within about 0.3 seconds. The two modes differ only in who carries out a decision that has already been approved. The step toward autonomy is therefore small and auditable rather than a change in kind, which reflects the principle guiding this project: autonomy is grown in supervised steps, not switched on.

4.9 Safety Architecture

In drive mode the computer commands the robot's movement, so safety is arranged in parts. No single failure should be able to produce movement that nobody asked for. Five independent mechanisms apply.

1. The robot moves only while the operator holds a key. Releasing it stops the robot within about

0.3 seconds.

2. A proximity guard watches the LiDAR and freezes the robot whenever anything enters a zone ahead of it, even if the key is still held. If the LiDAR data stops arriving for more than a second, the guard treats the path as blocked rather than clear.
3. The abort key stops the robot and exits, from any stage of the process.
4. The robot's onboard gateway expects a command at least every 250 milliseconds. If the workstation crashes, hangs, or loses its wireless connection, the robot stops on its own.
5. Speed limits are applied before any command is sent, and routes are planned only through free cells with a margin, so an unreachable or unsafe target is never proposed in the first place.

One limitation should be stated plainly. The proximity guard stops for people; it does not plan a way around them. Routing around obstacles is the job of a full navigation stack [40], which is the next stage of development. In a busy room, the mitigation is to use suggest-only mode instead.

4.10 Validation Methodology

No radiation source strong enough for algorithm development was available. A weak source would also have been of limited use, because it provides no known answer against which to score the estimate. The solution was to simulate the source instead.

A small program takes the place of the real detector. Once a second it reads the robot's actual position, calculates the dose rate that would be measured there, and sends that value to the recorder. The calculation is the physics of the situation: a background level, plus a contribution from each source that falls off with the square of the distance, with random variation applied to reflect the statistics of counting. The value is converted using the same calibration as the real detector and sent in the same format over the same connection. The recorder reads only the values, so nothing further down the chain can tell the difference. Switching back to the real detector is a single change on the command line.

Two features make this more than a convenience. First, the true source position is held in a file that no part of the estimating code reads. This was checked by inspecting which files import which: the model and the planner import only the shared calibration constant, never the source definitions. The estimate therefore cannot be influenced by the answer, even accidentally. Second, the readings depend on where the robot actually goes. A pre-recorded dataset cannot test a planner, because it cannot say what would have been measured somewhere the robot did not visit. Simulated fields of this kind are the accepted standard for developing planning algorithms [35, 36]. The novelty here is that only the radiation is simulated; everything else is real.

4.11 Experimental Design

Development and rehearsal took place in a university laboratory. It is a cluttered and occupied space, with furniture at the height the sensor scans, which makes it deliberately harder for scan matching than the rooms the system is intended for. The scored surveys reported in Chapter 5 took place in the rooms described in Section 5.2.

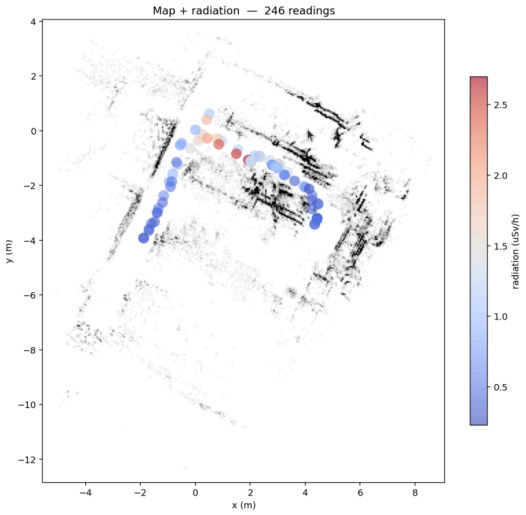

Figure 8 A survey run against a simulated source. Everything here is real except the radiation itself: the robot, the room, the position estimate and the recorded positions are all genuine, while the radiation values are calculated from the live position against a source the estimating code cannot read. The peak reaches roughly 2.5 μSv/h, far stronger than the real sources used later, which is what made development practical.

4.11.1 Structure of the field campaign

Access to radioactive sources and to a suitable room was intermittent, so the work ran on two tracks. Rehearsals using the simulated source ran continuously in the laboratory throughout development. Every component, and every repair made afterwards, was exercised and scored there before any session with real material. Sessions in a test room were correspondingly rare, and were used only for definite verification.

Four sessions were held in total. The first checked our system from end to end using the real detector but without the intelligence layer: a map was built, readings were paired with positions, and the combined display was confirmed to work on real hardware. The second was run by a collaborator to test an alternative mapping method. It is not part of this report's claims, although its output is the intended replacement for the position estimate used here. The third was the first check against real radioactive material with the complete system. The fourth was the final survey, carried out after the faults seen in the third had been diagnosed and fixed. A scored simulation survey was also run on the same day as the third session, and provides the rehearsal figure reported alongside the two real results in Chapter 5.

The three scored surveys are therefore not three attempts at the same experiment. They are a rehearsal, followed by two field sessions separated by a round of repairs, and Chapter 5 treats them accordingly.

Drift was to be measured by a loop test. The robot's starting position is marked on the floor, the robot is driven a loop of known length back to that mark, and the position it reports at the end is compared with the position it started from. The distance between the two is the drift. Because the map builder works with any source of position, the same loop can be repeated with each method and the results compared directly.

The intelligence layer is evaluated against known source positions. Four measures are used. The first

is the distance from the model's highest point to the true source, compared with published results of 0.09 to 0.59 m on real hardware at room scale [38]. The second is how the error in the estimated field falls as readings accumulate, which shows when further measurement stops helping, benchmarked against convergence in roughly fifteen readings [12]. The third is the number of steps before the stopping rule fires. Final one, is how much error is removed per metre walked.

4.12 Implementation and Verification Approach

The intelligence layer was built test-first, across fourteen tasks: nine for the survey logic and five for autonomous movement. Each task was independently reviewed before the next began, with a review of the whole branch at the end.

The components were separated so that all the decision logic is pure computation. The field model, the scoring rule, the timing rule and the stopping rule depend only on arrays and the map. None of them needs a robot, a network connection, or a running system to be tested. Interaction with hardware is confined to thin layers at the edges. This separation is what made the testing described below possible at all.

4.12.1 Test suite

The suite contained 72 tests when the build was finished, and grew to 94 during the field campaign as each fault found in operation was secured by a new test. It covers the physics model, including the inverse square calculation, the counting statistics and the calibration; the mathematics of the Gaussian process, checked independently against a reference implementation and agreeing to within one part in a million; the geometry of the map, including image orientation, behaviour at the boundaries, and the safety margins; the scoring, timing and stopping rules, including the case where the data is already dense enough that the survey should stop; route planning on test maps, including movements around walls and the case where no route exists; the proximity guard, including its behaviour when sensor data goes stale; the operator key handling at every stage; and the guarantee that the motion controller sends a stop command on every path out of the code, including when an error occurs.

Chapter 5: Results and Analysis

5.1 Evidence Tiers

Results were obtained at four different levels of realism, and every claim in this chapter is labelled with the level at which it was established (Table 3). The distinction matters. A result obtained in software says nothing about how the hardware behaves. A result obtained with a simulated radiation field says nothing about how the detector responds.

Table 3 Levels of evidence used in this chapter.

Level	Name	What is real
T1	Offline replay	A recorded map and survey log. No robot, no movement
T2	Backtest	A recorded map, the real command path and the real decision logic. Movement is simulated
T3	On robot, simulated source	Robot, room, position estimate, movement, timing and logging are all real. Only the radiation is simulated
T4	On robot, real source	Nothing is simulated

This progression follows accepted practice, in which simulated fields support the development of planning algorithms while the strongest claims are reserved for the highest level of realism [35, 36]. Earlier studies using simulated sources ran entirely in simulation. Here, at level T3, only the radiation is simulated. Results at level T4 were obtained, and are reported in Sections 5.5 and 5.7.

5.2 Live Survey Performance (T3–T4)

The pipeline ran from end to end throughout the campaign. Each session accumulated on the order of a hundred thousand LiDAR points, and the map built on the final day used 800 scans and 546,744 points. Manual control was unaffected at all times, which is what an architecture with no command channel should have. The three scored surveys ran for 5.0, 7.5 and 8.2 minutes, producing 300, 450 and 495 readings paired with positions. Each was started with a single command, and all outputs were cleared at the start so that no run could display data from an earlier one.

Two rooms were used. Development and all simulation work took place in a university laboratory, in a section of it cleared for testing, which produced a map covering 16.50×16.60 m. Both surveys with real sources took place in the radiation room where the radioactive was located. That room was mapped separately on each of the two days, producing maps covering 5.10×11.30 m and 4.80×9.35 m. These figures are the extents of the maps the system produced rather than the dimensions of the rooms themselves, since a map covers only the ground the robot drove.

5.3 Mapping Results by Position Source (T3)

Using leg odometry alone produced clean maps that clearly resembled the room. Their only issue was drift: structure the robot returned to was drawn slightly out of place. Flattening the scans and passing them to slam_toolbox produced a much worse result, a blurred shape rather than a room, because projecting a sparse 3D scan into a thin flat one throws away most of what scan matching needs. KISS-ICP with its default settings produced unstable position estimates and worsen the map;

once its settings were matched to the size of a room, tracking improved noticeably. No live method produced a map that was both clean and free of drift in a cluttered space. This is the practical reason for the two-stage approach used this system: build the map once, then survey against it.

5.4 Localization Drift (T3)

Drift was never measured. The loop test described in Section 4.11 was not carried out, and the recordings contain no comparison between a start and end position at the same place.

This is not only a gap in effort. The positions in the log are the robot's own estimate, so if that estimate has drifted, the log cannot show it. Detecting drift requires an independent reference that says where the robot really was, and no such reference was recorded. The only evidence available is an observation from the final session: with the robot standing at a wall, its displayed position lagged its physical position by roughly 0.2 m. That is an observation, not a measurement, and is reported as such.

Drift nevertheless affects every distance quoted in this chapter, so it is worth being clear about how it enters the error figures. The true source position was recorded by driving the robot next to the source and reading the position it reported, in the same session as the survey. The source position and the readings are therefore expressed in the same coordinate system, the one the robot established when it was switched on.

This matters because that coordinate system may be offset from the real world. If it is, everything within it is offset by the same amount, and the distance between the estimate and the recorded truth is unaffected. A map printed slightly off-centre still shows the correct distance between two towns. Drift that accumulates gradually cancels in the same way for most of a survey, since readings taken minutes before the truth was recorded share most of the same accumulated error.

What does not cancel is the drift that built up between taking the last reading and recording the source position. The errors reported below are therefore accurate within the survey's own coordinate system, and the 0.2 m observation above indicates the scale of what remains uncertain.

5.5 Measurement of a Real Radiation Field (T4)

5.5.1 The sources

The sources were zirconium-89 waste held in standard one-litre clinical sharps containers measuring $105 \times 100 \times 190$ mm. This is a clinically realistic choice rather than a laboratory convenience. Zirconium-89 is used in immuno-PET imaging, and its 78.4-hour half-life is chosen to match the behaviour of the antibodies it is attached to. The containers were genuine departmental waste of the kind a nuclear medicine facility produces routinely.

What the detector could see follows from how the isotope decays. Zirconium-89 decays to yttrium-89, by electron capture in 77% of cases and by positron emission in the remaining 23%. The resulting transition emits a 0.909 MeV gamma ray in 99.9% of decays, with three weaker lines above 1.6 MeV that together account for less than 1%. The positrons themselves travel only a few millimetres before meeting an electron, producing a pair of 0.511 MeV gamma rays. The field the robot mapped is therefore made of gamma rays, dominated by the 0.909 MeV line with a contribution at 0.511 MeV. The

charged particles are absorbed inside the container and contribute nothing at the distances involved in a survey.

An independent instrument gives a sense of the source strength. A Berthold LB124 contamination monitor held within 2 cm of a container recorded up to 1000 counts per second at its most active face. The total activity was not recorded.

Two points follow for the results below. First, a container of this size is not a point source. Its dimensions, between 0.1 and 0.19 m, are comparable to the localization errors reported in Section 5.7, so an estimate accurate to 0.12 m is locating the source to within its own size. Second, the inverse-square model used by both the field model and the simulated source treats a distributed volume of waste inside an absorbing plastic container as a single point. This approximation contributes to the remaining error, alongside counting noise and position error.

Before considering any estimate, it is worth establishing that a real signal was captured at all. In both real-source runs the detector recorded a background of 0.028 to 0.04 μSv/h, rising to between 0.09 and 0.11 μSv/h near a source. That is an increase of roughly three times: a genuinely weak signal, and exactly the low-count situation the noise model was built for.

The geometry supports treating the field as inverse-square. The detector is carried on the robot's body while the source stands on the floor, so the closest the two came during a survey is estimated at 10 to 40 cm. Over such distances the fall-off with distance dominates, and absorption of 0.909 MeV gamma rays in air is negligible. The offset $h = 0.30$ m used in the fits below represents this vertical separation. It was taken from the physical arrangement rather than fitted to the data.

The spatial pattern of the readings confirms the source. Fitting dose rate against distance using $v(d) = A/(d^2 + h^2) + b$ gives $R^2 = 0.742$ for run B and 0.391 for run C. The corresponding correlations are $r = 0.861$ over 450 readings and $r = 0.625$ over 495. Correlations of that strength over hundreds of independent readings cannot come from noise. The background levels the fits recover, 0.047 and 0.050 μSv/h, match the observed baselines. The readings also behave correctly in time: the dose rises above background only while the robot is close to the source, and falls again once it leaves (Figure 10).

Figure 12 shows the same readings laid out spatially, without any model applied. Taken with the fits above, this establishes that a real field was measured before any estimation was attempted. Run C's weaker fit reflects a fainter source: its peak stands only about 1.8 times its background, against 2.4 for run B. That makes the accuracy achieved in run C, reported next, the more demanding result.

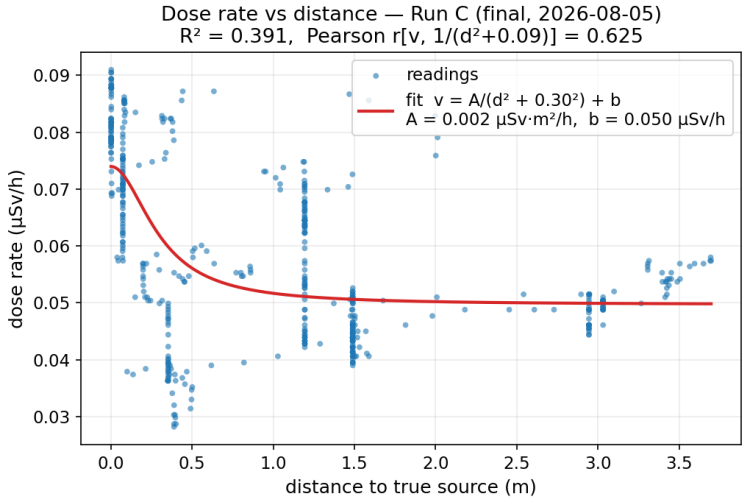

Figure 9 Dose rate against distance to the source, final run, with the fitted inverse-square curve. This fit is the main evidence that a real radiation field was measured rather than detector noise.

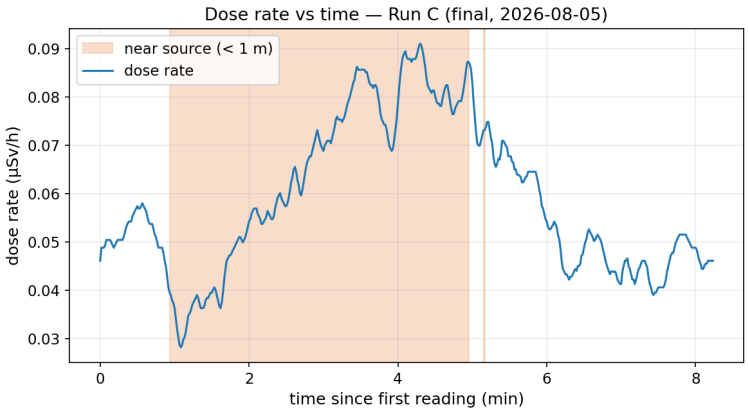

Figure 10 Radiation against time, final run. Readings rise above background only while the robot is within one metre of the source (shaded) and fall again once it leaves.

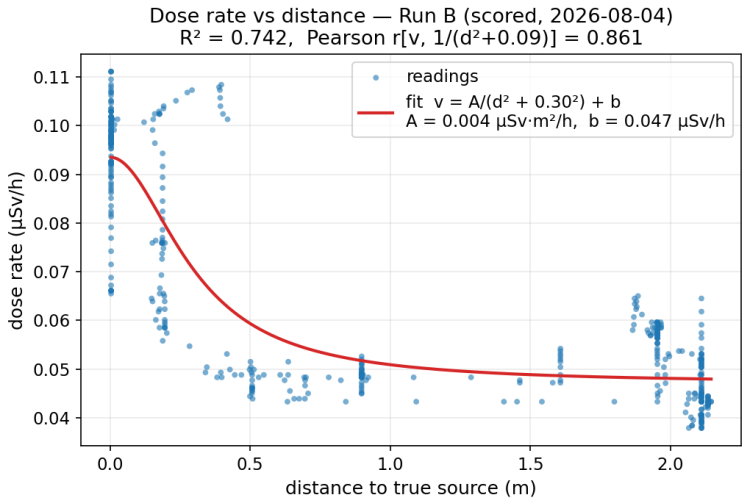

Figure 11 Radiation against distance for the first real-source survey. A stronger, clustered emission gives a tighter fit ($R^2 = 0.742$, $r = 0.861$). The contrast with Figure 9 reflects a real difference in signal strength, not a difference in method.

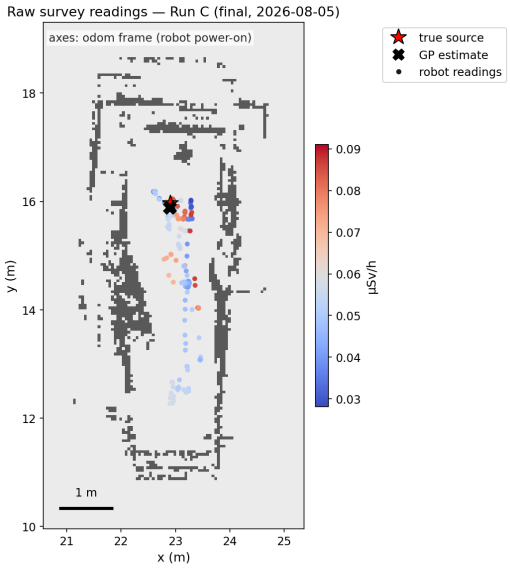

Figure 12 Raw readings from the final survey, coloured by dose rate, with no model applied. The higher readings cluster at the recorded source position without any interpolation, so the result in Section 5.7 is supported by the measurements themselves and not only by the model fitted to them.

5.6 Verification of the Decision Logic (T1–T2)

The decision logic was tested in software, both before the campaign and throughout it. Ninety-four automated tests, spread across sixteen files, cover the physics model and the mathematics of the field model. They cover the geometry of the map, and the rules for choosing where to measure, how long to measure, and when to stop. They cover route planning, including the case where no route exists at all. They also cover the proximity guard when sensor data stops arriving, the handling of the operator's keys, and the guarantee that the controller always sends a stop command before it exits.

Independent review during the original build found three defects that mattered for safety. The first was in the check for a clear path, which sampled only nine points along the route. A wall thinner than the gaps between those points could pass unnoticed. The second was in the movement routine, which could exit without sending a stop command if an error occurred early. The third was a confirmation prompt that quietly treated the abort key as "skip" at one stage of the process. The field campaign then found eleven further faults, listed with their causes and fixes in Table 4.

Each fix was secured by a test where one could be written, which is why the suite grew from 72 tests to 94 during the campaign. The whole system was then re-checked against five simulated driving scenarios before returning to the robot, and passed all five. A clear-corridor drive finished 0.14 m from its target. A detour around a mapped obstacle took 5.1 m against a 2.0 m straight line. A simulated person stepped into the path and blocked it for five seconds; the robot froze and stayed frozen for 6.8 s, slightly longer than the blockage because obstacles are remembered for a short time after they disappear, then resumed once clear. A six-checkpoint survey over 134 readings estimated the source position to within 0.25 m. And a drive through the genuine command path, rather than a stand-in, arrived within 0.13 m with the stop command confirmed as delivered.

Three of the eleven faults could not be secured by a test, and this is stated plainly. The bridge fixes lie in inherited code for which no test framework exists, and the detector wiring fault is a procedural problem rather than a software one.

Table 4 Faults found during the field campaign, with their causes and fixes.

What was observed	Cause	Fix
Position frozen for every reader	The bridge disconnected read-only clients after three seconds	Timeout applied only to clients sending commands; stale data now reported clearly
Detector reading zero or going silent	Wiring and USB connections failing under the vibration of a walking robot	Bench check before each session; connectors secured
Robot walked backwards or sideways	Control allowed sideways movement	Turn to face the target first, then drive forwards only
Very slow progress	The controller aimed at waypoints 0.3 m apart and never reached full speed	Travel at full speed between waypoints, slowing only on the final approach
Robot stuck at targets beside furniture	The 0.6 m safety zone makes such targets unreachable by rule	Measure where it stands if within 0.75 m of the target
Robot aimed underneath a bench	Thin legs appear and disappear between scans	Remember obstacles for three seconds; 0.7 s for the stop zone
Steering around nothing on clear ground	Memory accumulated stray returns and the robot's own legs	Require three remembered points; ignore returns within 0.20 m
Rocking back and forth when hemmed in	Six seconds of patience, repeatedly reset by flicker	Replan after 1.5 s without progress
Proposals in unsurveyed areas	Uncertainty is highest where nothing was measured	Restrict proposals to the measured area plus 0.6 m
Bridge process died mid-run	Replying to a client that had vanished raised an unhandled error	Drop that client, stop safely, keep serving the others
Marker mistaken for a source claim	Proposal and estimate looked alike	Separate marks for the two

5.7 Source Localization Results (T3–T4)

The central result of this project is that the robot carried out the full cycle it was designed for: it built a model of the radiation field from its own readings, chose where to measure next, planned a route to that point, walked there under supervision, measured, and updated the model. The accuracy with which it finally located the source is how that capability is scored, and it is reported here as the headline figure. Table 5 gives the three scored surveys.

Table 5 Scored surveys. The error is the distance from the model's highest point to the independently recorded source position.

Run	Detector	Readings	Duration (min)	Room (m)	Error (m)
A, simulation rehearsal	Simulated (T3)	300	5.0	16.5×16.6	0.29
B, real source, first	Real J305 (T4)	450	7.5	5.1×11.3	0.36
C, real source, final	Real J305 (T4)	495	8.2	4.8×9.4	**0.12**

Some figures in this chapter carry a date in their title. These are the dates on which the recorded data was scored, which in each case is the day after the survey itself. In both real runs the scored source stood near the middle of the surveyed corridor. In the first run the estimate was 0.22 m off across the corridor and 0.28 m off along it. In the final part the same figures were 0.03 m and 0.11 m, a displacement comparable to the width of the container itself.

The final run located a real radioactive source to within 0.12 m (Figure 13). For comparison, published results using real hardware at room scale report 0.09 to 0.59 m [38], and a particle-filter method evaluated in simulation reports 0.35 to 0.5 m [13]. This result therefore sits at the better end of the published hardware range and below the simulation-only figures. It was achieved with an uncalibrated Geiger–Müller tube costing a few tens of pounds, an uncorrected position estimate, and a source whose signal was only about three times background.

Two observations qualify this result and strengthen it in equal measure.

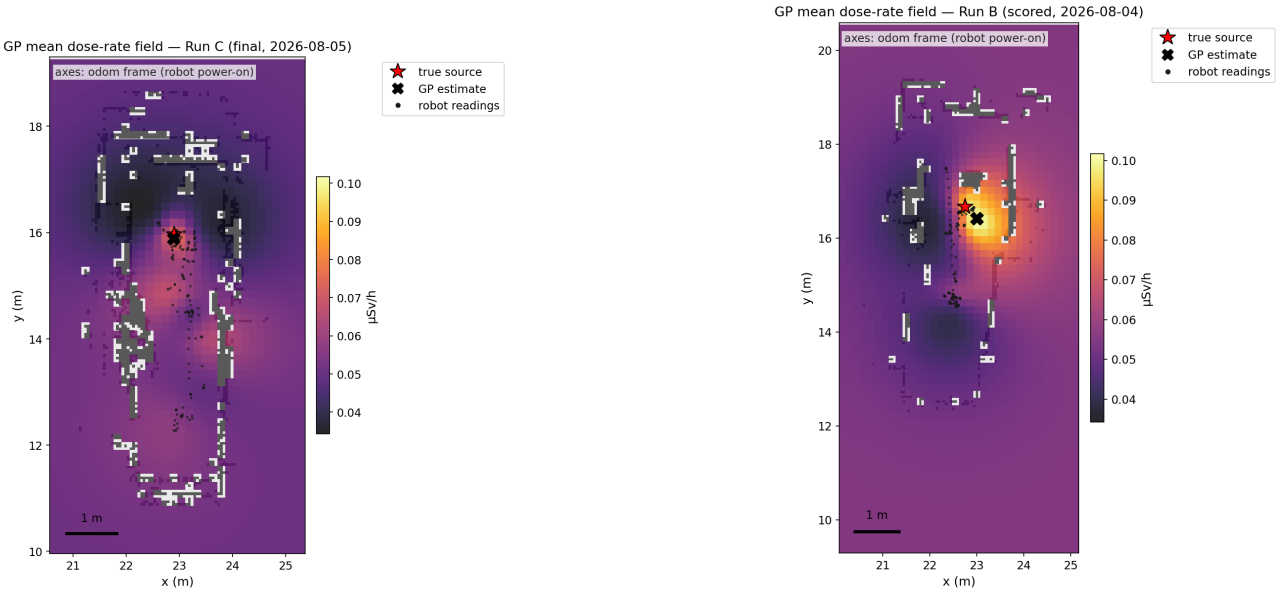

Figure 13 Estimated Radiation fields for the two real-source surveys, drawn on the same colour scale. Left: the final survey (495 readings), where the estimated source position ($\times$) lies 0.12 m from the recorded one ($\star$). Right: the first survey (450 readings), scored at 0.36 m. The right-hand field looks brighter and more sharply peaked because three sources were present and their combined emission was stronger. The left-hand field is fainter because its single source was weak. The difference in appearance therefore reflects signal strength rather than the quality of the estimate: the fainter field gave the more accurate answer.

First, the estimate does not depend on how measurements were chosen. It is simply the highest point of the fitted field, and the rule that selects the next measurement plays no part in it. Re-running the recorded data with the two alternative selection rules gives an identical estimate and an identical error of 0.117 m. What the selection rule determines is where the robot goes next, not what it concludes from what it already has.

Second, and more importantly, the accuracy survived a serious operational failure. The final run's autonomous movement was progressively destroyed by a failing wireless link. The command connection cycled through connecting, authenticating and dropping roughly thirty times, with the robot stopping itself at each break, and the walk ended when the position feed was lost after 2.18 m of travel (Figure 14). Despite this, the survey produced the best result of the campaign, because the estimate is drawn from all the readings collected rather than from any completed walk. The system separates gathering data from reaching conclusions, so a disrupted walk costs coverage and convenience but not correctness. That is a property worth designing for deliberately in field robotics, where equipment failure is normal rather than exceptional.

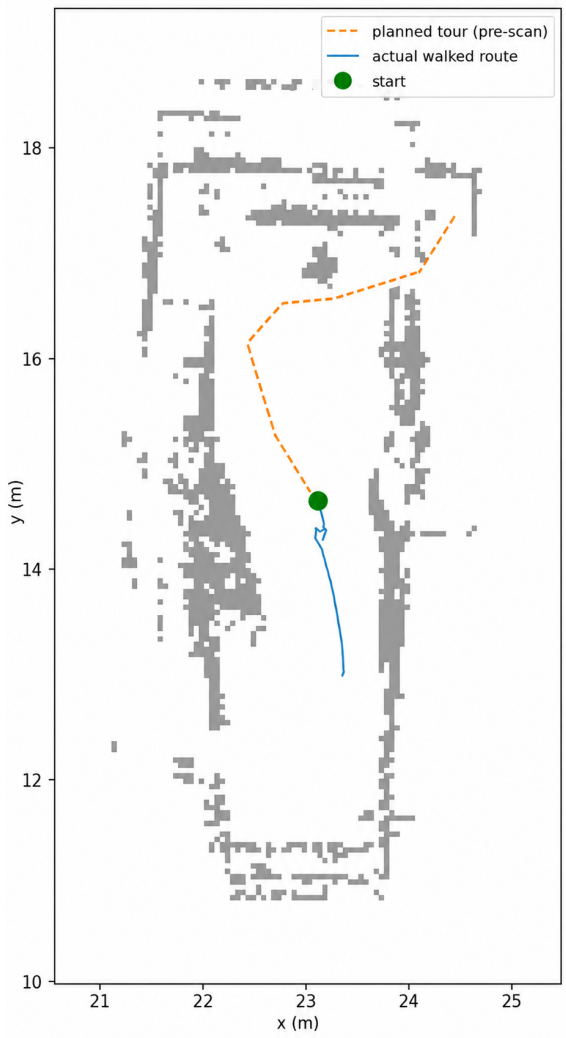

Figure 14 Planned route against the route actually taken in the final survey. The plan is shown alongside the path walked before the wireless failure ended the drive after 2.18 m. The gap between them is the visual counterpart of the argument in the text: the survey's conclusions come from the readings collected, not from finishing the plan.

5.8 Why the Two Real-Source Runs Differed

The first real-source survey located its target to 0.36 m; the second reached 0.12 m. The obvious explanation is that the system was faulty in the first run and repaired before the second. The found data does not support that, and the actual explanation is more interesting.

Four possible causes were tested by re-running the analysis on the recorded data from both runs.

The estimation code was not the cause. Re-scoring run B with the current code gives 0.357 m, the same figure it produced at the time. Nothing in the estimation path has altered since before either run, so the improvement cannot be attributed to it.

The restriction on proposals was not the cause. That rule governs where the robot proposes to go next. It plays no part in producing the final estimate, which is simply the highest point of the fitted field. Applying the rule to the estimate changes nothing in either run.

The extra readings were not the cause. Run C collected 45 more readings than run B. However, the model's uncertainty at the true source position stops improving after roughly 300 readings in both runs. The additional readings therefore added no information where it mattered.

Coverage made some difference. Run C's readings were spread over 2.34 m^2 against run B's 1.25 m^2, and 225 of them fell within half a metre of the source, against 178 in run B. Better coverage of the region around the source helps, but it does not account for the whole difference.

The test conditions were not the same, and this is the largest factor. Run B was carried out with three sources in the room, placed close together, but the result was scored against only the strongest of them. A model fitted to the combined emission of several nearby sources will place its peak somewhere between them, weighted toward the strongest. That is the model behaving correctly, not failing: the field really does peak there. The error figure is misleading because it compares that peak against one source rather than against the group. Modelling plausible arrangements of three sources suggests the companions could pull the estimate by up to roughly 0.13 m.

That research also revealed something more important, and it changes how the headline figure should be read. A single source was simulated at exactly the recorded position, sampled at the same positions run B actually measured from, and fitted with the same model. The estimate came out 0.124 m away from the truth. In other words, the method has a built-in inaccuracy of that size under these conditions, even when everything else is perfect. The cause is that the model smooths over a scale of about a metre, and the readings were collected along a narrow path rather than spread around the source, so the fitted peak is pulled toward where the readings happen to be.

The conclusion therefore has three parts. Run B's 0.356 m is made up of roughly 0.12 m that the method itself contributes, up to 0.13 m from the additional sources, and the remainder from counting noise, position error and imperfect estimation of the background. Run C's 0.12 m is the accuracy of the method with a single source. And because that figure is close to the method's own built-in limit at this density of readings, the constraint is no longer the detector or the statistics. It is the smoothing scale of the model and the geometry of where the robot happened to measure. Improving further would mean changing those, not buying a better detector.

Finally, the model's sensitivity to its settings should be reported rather than hidden. The length scale

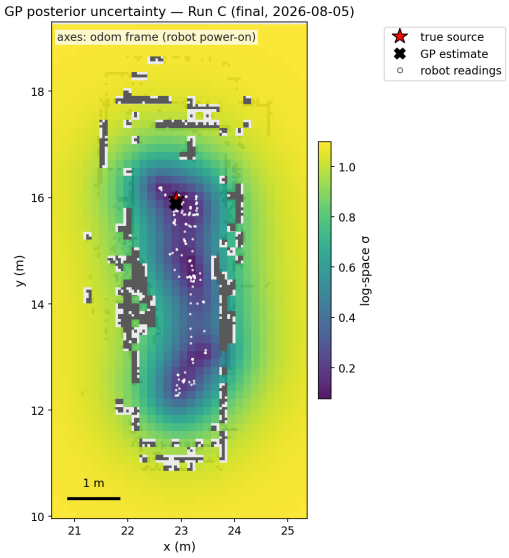

Figure 15 The model's uncertainty across the final survey. Confidence follows the data: uncertainty is lowest along the route the robot actually took, and rises with distance from any reading. This is the quantity that drives the robot toward unexplored ground.

controls how far one reading influences the estimate at neighbouring points. Varying it between 0.5 and 2.0 m gives errors of 0.357, 0.357 and 0.827 m for run B, and 2.061, 0.117 and 0.117 m for run C. Each run tolerates a change by a factor of two, but only in one direction, and the value used throughout, 1.0 m, is the only setting that works for both. The result is therefore robust within limits, and this sensitivity belongs alongside the headline figure rather than in a footnote.

5.9 Model Validation (T4 data)

Two questions decide whether the model deserves the confidence placed in it. Does it predict readings it has not seen? And when it says how uncertain it is, is that honest?

The first question is tested by hiding one reading, predicting it from all the others, and comparing the prediction against the reading that was hidden. Repeating this for every reading gives an average error of 0.0084 μSv/h for run B and 0.0100 μSv/h for run C, which is 13.2% and 17.0% of the average signal. This is worth comparing against the readings themselves. A single reading at these count rates carries 30 to 50% noise. The model therefore predicts a reading it has never seen more accurately than that reading measures itself. This is what combining information across many nearby readings is supposed to achieve, and here it is quantified.

The second question gives a less flattering answer, and it should be stated plainly. When the model reports an uncertainty, it is claiming that the true value lies within one standard deviation of its prediction about 68% of the time. Testing this on the hidden readings, the true value fell inside that range 62.7% of the time in run B and 53.3% in run C. The model is therefore slightly more confident than it has earned. Widening the range to include the detector's own measurement noise pushes this to 100%, which is too generous in the other direction, because that noise allowance is larger than the errors actually observed.

The honest claim is that the uncertainty is right in shape but not in scale. It is low where readings are dense, high where they are absent, and it stops falling at the source once enough readings have

been taken (Figure 15). It is not calibrated as a number. The noise allowance is the setting that would correct this. Because uncertainty is what draws the robot toward unmeasured ground, this affects where the robot chooses to go next, but not the source position it finally reports.

5.10 Behaviour When No Source Is Present

A system that finds sources must also be willing to report that there is nothing to find. No background-only survey was recorded in the field, so this was tested offline instead. A simulated dataset of 120 background readings was generated over the final run's real map, sampled at 0.03 μSv/h through the real calibration. At that rate, 46 of the 120 readings came out as exactly zero, which is what a real detector does at background.

The stopping rule behaved exactly as intended. Its measure came out at 0.0128, already below the 0.02 threshold at the very first proposal. It stayed there, and on the third attempt the planner declared the survey finished. Given a field with nothing worth measuring, the system stops walking rather than wandering.

The rule that marks a suspected source, however, failed, and it failed in both directions. That rule marks a source only where the highest point of the field is more than 1.5 times the median value across the map. On the background-only data it passed, and marked three sources that did not exist. The reason is that so many readings were exactly zero that the median was dragged down close to zero, which makes exceeding it by half again very easy. On run C's real data, which did contain a source, the highest point reached 1.4996 times the median: the rule failed by three parts in ten thousand, and no source was marked at all. Comparing a peak against a median is simply the wrong test when most of the data is counting noise.

A redesign tested offline fixes this. Instead of comparing against the median by a ratio, a candidate must exceed the median by at least four standard deviations, which asks whether the peak is large relative to the scatter in the data rather than relative to its middle value. Peaks are then selected one at a time and kept at least 1.5 m apart. On the background-only data this gives no peaks at all. On run C it gives exactly one, 0.063 m from the true source. On run B it gives one at 0.141 m. In both real cases this is closer to the truth than the unconstrained maximum, because the misleading candidate in run B fails the test and is discarded. Any threshold between roughly 3.4 and 4.7 standard deviations works, so the chosen value of four sits comfortably in the middle.

This replacement has been tested on the recorded data but was not part of the code that ran the campaign. It is presented as such: a flaw found by auditing the system after the fact, with a remedy designed and demonstrated but not yet deployed. It does not affect the headline results, which are produced by a separate part of the code.

5.11 Evaluation Against Objectives

Table 6 assesses the objectives set out in Section 3.5 against the evidence obtained.

Two objectives are considered as only partly met. Drift was described but never measured, for the reason given in Section 5.4. And although the robot planned routes, walked them under supervision, steered around obstacles that were not on the map, and measured at positions it chose itself, no

Table 6 Objectives against evidence.

	Objective	Outcome	Evidence
O1	Read data live without disturbing manual control	Met	Three campaign sessions with no conflict observed (§5.2)
O2	Live floor plan with selectable position source	Met	Maps built in three rooms; position source interchangeable (§5.3)
O3	Radiation paired with position, aligned to the map	Met	300, 450 and 495 paired readings; combined displays (§5.2, §5.5)
O4	Measure drift and evaluate corrections	Partly met	Methods compared by inspection (§5.3); drift never measured (§5.4)
O5	Field estimate, cost-aware selection, timing and stopping rules	Met	Implemented and verified (§5.6); stopping shown offline (§5.10)
O6	Autonomous movement under supervision, with layered safety	Partly met	Waypoints walked under the hold gate; no survey finished on its own criterion (§5.7)
O7	Validate against known positions, reporting error in metres	Met	0.29, 0.36 and 0.12 m against recorded truth (§5.7)

survey ever ran through to its own stopping criterion in the field. Every run was ended by the operator or by an equipment failure. The stopping rule is shown to work offline; its behaviour in the field is not established.

5.12 What Is Novel and What Is Engineering

Table 7 separates what is claimed as novel from what is competent engineering.

Table 7 Novel contributions distinguished from engineering work.

Component	Status
Operator approval of each individual measurement, refined to a held key during movement	Novel and unpublished. The closest work switches between levels of autonomy [32] or chooses points on its own [12]
Measurement duration derived from counting statistics for a Geiger–Müller tube on a robot	Novel; the literature uses fixed durations [26]
Scored validation using a fake(simulated) source on a real robot in a real room	A contribution of method; earlier studies with simulated sources ran wholly in simulation [35, 36]
Replacing the source-marking rule with a significance test	A contribution at the level of analysis; validated offline, not deployed (§5.10)
Comparison of selection rules on an indoor task constrained by a map	Partly addressed; the estimate does not depend on the rule, and a full comparison needs live runs
The field model, the selection rule, the route planner and the stopping rule individually	Established methods, correctly cited. The contribution is combining them into one supervised system validated on hardware
The held-key gate, the proximity guard, fault recovery and bridge hardening	Engineering, but reviewed and regression-tested engineering, which is itself evidence of process

5.13 Limitations

The limitations are given in order of how much they matter in this project.

Amount of evidence. Two surveys with a real source is a small sample. No runs were repeated from different starting positions, no control run without a source was carried out in the field, and the comparison of selection rules could only be done offline, where it cannot distinguish the rules by how the survey actually behaves. The offline substitutes reported above are honest, but weaker than repeating the work in the field.

The stopping rule was never used in the field. Every run ended because the operator stopped it or because equipment failed.

Drift was never measured. The figure quoted is an observation, and the recordings cannot in principle reveal their own drift.

Run B's conditions were not controlled. Three sources were present and one was scored. Their relative positions were never recorded, so the effect of the others can be considered as modeled but not calculated.

The source model is an approximation. The emitter is a volume of waste inside an absorbing container, treated throughout as a single point obeying an inverse-square law. This is reasonable at survey distances but contributes systematically to the remaining error, and the container's own size is comparable to the accuracy achieved.

The instrument is not calibrated. The detector is a hobbyist assembly, and its conversion factor is a manufacturer's figure with a provisional correction applied, not a traceable calibration. No claim of clinical measurement accuracy is made. Its wiring also failed repeatedly under the vibration of a walking robot.

The uncertainty is not numerically calibrated, as Section 5.9 quantifies, and the accuracy reported is partly limited by a bias of about 0.12 m inherent to the method at the densities of readings achieved.

Infrastructure. A failing wireless link cut short the final drive. The command protocol deliberately treats such thing as a lost connection and stops the robot, which is the right choice for a robot working near radiation, but it makes the whole process dependent on the quality of the link.

Map accuracy at the edges. A cell is recorded as free wherever nothing was detected, which near the limits of a survey overstates how much space is actually traversable. Combined with a path check that treats the robot as a point rather than a body, this allowed targets slightly beyond the true boundary to be accepted, and the robot to be sent toward positions it could not physically get to. Restricting proposals to the measured area removes the symptom. Modelling the robot's size, and distinguishing space that was never observed from space observed to be empty, would address the cause.

Scope. The proximity guard stops for people but does not plan a way around them. The map and the field are two-dimensional. The multi-point tour planner was built and tested in simulation but never used in a real survey. And the environments were small rooms containing one or three sources.

Chapter 6: Conclusions and Future Work

6.1 Conclusions

This project set out to move a hospital radiation-survey robot from recording radiation to reasoning about it, and to build the foundations that change requires.

The first research question asked whether a survey could be carried out live, in one run, without interfering with manual control. It can. The system reads the robot's sensor streams without ever sending a command, which allows the floor plan and the radiation map to be built while the operator

drives. This replaces a system that previously took three separate stages. It was demonstrated across three sessions, producing 300, 450 and 495 readings paired with positions.

The second question asked how the robot's position estimate drifts, and whether the available methods can correct it. The answer is partial, and worth stating precisely. Leg odometry produces clean maps that drift slowly. Flattening the 3D scan into a flat one throws away too much for two-dimensional SLAM to be useful. 3D scan matching helps, but only when its settings are matched to the size of a room. The reliable answer turned out to be architectural rather than a choice of tool: build the map once, then locate the robot within that fixed map. The mapping code was made independent of the position source, so any method can be substituted. Drift itself, however, was never studied in depth, and that gap limits every distance quoted in this report.

The third question asked whether the robot could plan and carry out its own measurements. It can. A statistical model estimates the radiation field and its own uncertainty. A scoring rule chooses each next measurement, weighing what would be learned against the distance to walk. The time spent at each point follows from counting statistics rather than convention, and the survey ends on statistical proof rather than a fixed frame. Against a real radioactive source the system located the hotspot to within 0.12 m. That is inside the 0.09 to 0.59 m range reported for published systems using real hardware, and better than results from simulation on its own. It was achieved with an uncalibrated detector costing a few tens of pounds, an uncorrected position estimate, and a source only about three times stronger than background. Analysis afterwards showed this figure sits close to the method's own built-in limit at that density of readings, which places the remaining constraint in the model's smoothing and the geometry of the survey rather than in the detector or the statistics.

Three further findings deserve emphasis. First, accuracy survived operational failure: the best result of the campaign came from a run whose autonomous movement collapsed when the wireless link failed, because the conclusions are drawn from the readings collected rather than from finishing a route. Second, the first real-source run, which looks worse at 0.36 m, is better understood as a single-source measure applied to a field containing three sources, of which roughly 0.12 m is bias the method produces regardless. Third, auditing the rule that marks a suspected source found it failing in both directions: it invented sources on background data, and missed a genuine one by three parts in ten thousand. A replacement based on a significance test was designed and tested on the recorded data, and improves the result on both real runs.

The principle guiding this project is that autonomy is grown in supervised steps. Each stage moves initiative toward the machine while leaving the human able to stop it: taught routes, then a live survey driven by hand, then the robot proposing and the human approving, then the robot executing while the human holds a key. What remains before full autonomy is the removal of that key, not the invention of the capability behind it.

6.2 Impact

Technically, the platform is now an extensible system rather than a fixed tool. The source of position information can be swapped without touching anything else. The passive bridge supplies live data to any future component without interfering with control. And the survey log and map data define the interfaces that other components can build against.

The negative results are also useful. Network routes that turned out to be unavailable, the failure of flattening scans for two-dimensional mapping, and a default setting that quietly ruins indoor performance are all documented, which saves the next person from repeating them. The verification record offers a starting point in terms of safety for future work.

For the clinical goal, the change is concrete. A survey becomes one live run rather than three staged ones. Radiation feedback arrives during the survey rather than after it. Effort concentrates where the readings are informative. And the survey can say when it is finished. The wider point is that the accuracy came from principled statistics rather than expensive equipment: a hobbyist detector on an affordable robot matched results published using laboratory-grade instruments. If robotic radiation survey is to reach ordinary hospitals rather than national laboratories, that is the direction the argument has to run.

The limits of the work should also be stated clearly. This is a research prototype. Its dose figures are provisional estimates from an uncalibrated detector, and no claim is made about clinical measurement accuracy. Deploying it near patients would require calibration against traceable standards, approval under institutional radiation-safety governance, and a formal risk assessment. The supervision architecture was kept deliberately rather than as a shortcoming, following evidence that operators of robots in radiological work need behaviour they can interrupt and understand.

6.3 Future Work

The most immediate work is experimental rather than technical. Repeating the survey from different starting positions would replace two results with a distribution. Running a control system with no source present, and comparing the selection rules on the robot rather than offline, would turn two offline substitutes into field evidence. Carrying out the loop test would finally attach a number to the drift that limits every distance in this report. And the replacement source-marking rule, already designed and tested on recorded data, should be deployed and checked in the field.

The remaining improvements follow directly from the limitations. Continuous map-based localization would remove the drift limit, and would allow a question the literature itself identifies as unexamined to be answered: how much of the error in locating a source comes from error in knowing where the robot is. A full navigation stack [40] would let the robot plan around people rather than wait for them making it more complex. Frontier-based exploration [41] would remove the need for a map made in advance. Fitting the physics of the source directly to the survey log [37, 38] would produce an explicit estimate of source position alongside the field model, and would handle rooms containing several sources, as in the first real-source run. A more reliable command link, or moving the operator software onto the robot itself, would prevent the failure that cut short the final drive.

The final step is to remove the human gate for routine surveys in rooms that have been cleared, while keeping it wherever people are present, and to validate the system in a working clinical environment against a professional dosimeter, following the protocol used by the system's predecessor [2]. That would carry this work from a laboratory demonstration toward clinical use.

References

[1] UK Government, “The ionising radiations regulations 2017 (SI 2017/1075),” UK Statutory Instruments, 2017. [Online]. Available: https://www.legislation.gov.uk/uksi/2017/1075

[2] W.-C. Huang, B. Sendagorta Yanci, X. Wang, Q. Zeng, and S. Ni, “Development and evaluation of radiation monitoring using automated mechatronics system (quadruped),” MEng Group Project Portfolio, Department of Engineering, King’s College London, 2026.

[3] T. H. Lai, “Development and evaluation of a robotic radiation monitoring system,” BEng Research Project Report, King’s College London, 2026.

[4] A. Shafe, S. M. J. Mortazavi, A. Joharnia, and G. H. Safaeyan, “Development of RadRob15, a robot for detecting radioactive contamination in nuclear medicine departments,” *Journal of Biomedical Physics and Engineering*, vol. 6, no. 3, pp. 201–204, 2016.

[5] T. Winiarski, D. Gieldowski, K. Gieldowski, M. Tulik, M. Kobylecka, and J. Kunikowska, “Concepts for the use of assistive robots and artificial intelligence in a nuclear medicine facility,” *Clinical and Translational Imaging*, vol. 13, pp. 457–466, 2025.

[6] B. Nouri Rahmat Abadi, A. West, H. Peel, M. Nancekievill, C. Ballard, B. Lennox, O. Marjanovic, and K. Groves, “CARMA II: A ground vehicle for autonomous surveying of alpha, beta and gamma radiation,” *Frontiers in Robotics and AI*, vol. 10, p. 1137750, 2023.

[7] S. Sattar, M. J. Alam, Y. Mawla, M. A. Rahman, and M. A. Al Mamun, “Design and development of a wireless robotic system for radiation detection and measurement,” *Australian Journal of Engineering and Innovative Technology*, vol. 3, no. 4, pp. 57–63, 2021.

[8] P. Gabrlik, T. Lazna, T. Jilek, P. Sladek, and L. Zalud, “An automated heterogeneous robotic system for radiation surveys: Design and field testing,” *Journal of Field Robotics*, vol. 38, no. 6, pp. 657–683, 2021.

[9] Lawrence Berkeley National Laboratory, “Robotic gamma-ray imaging with a Boston Dynamics Spot quadruped,” 2023.

[10] CERN, “Quadruped robot (Unitree Go1) radiation inspection in the LHC tunnels,” 2023.

[11] A. West, I. Tsitsimpelis, M. Licata, A. Jazbec, L. Snoj, M. J. Joyce, and B. Lennox, “Use of Gaussian process regression for radiation mapping of a nuclear reactor with a mobile robot,” *Scientific Reports*, vol. 11, p. 13975, 2021.

[12] J. Adams, B. Cintas, N. Sung, A. Abrahao, L. Lagos, and D. McDaniel, “A behavioral robotics approach to radiation mapping using adaptive sampling,” *Applied Sciences*, vol. 15, no. 4, p. 2050, 2025.

[13] T. Lazna and L. Zalud, “Localizing multiple radiation sources actively with a particle filter,” arXiv:2305.15240, 2023.

[14] D. Shepard, “A two-dimensional interpolation function for irregularly-spaced data,” in *Proceedings of the 1968 23rd ACM National Conference*, 1968, pp. 517–524.

[15] H. Durrant-Whyte and T. Bailey, “Simultaneous localization and mapping: Part I,” *IEEE Robotics & Automation Magazine*, vol. 13, no. 2, pp. 99–110, 2006.

[16] M. Camurri, M. Ramezani, S. Nobili, and M. Fallon, “Pronto: A multi-sensor state estimator for legged robots in real-world scenarios,” *Frontiers in Robotics and AI*, vol. 7, p. 68, 2020.

[17] M. F. Fallon, M. Antone, N. Roy, and S. Teller, “Drift-free humanoid state estimation fusing kinematic, inertial and LiDAR sensing,” in *2014 IEEE-RAS International Conference on Humanoid Robots*, 2014, pp. 112–119.

[18] T. Bailey and H. Durrant-Whyte, “Simultaneous localization and mapping (SLAM): Part II,” *IEEE Robotics & Automation Magazine*, vol. 13, no. 3, pp. 108–117, 2006.

[19] S. Macenski and I. Jambrecic, “SLAM Toolbox: SLAM for the dynamic world,” *Journal of Open Source Software*, vol. 6, no. 61, p. 2783, 2021.

[20] 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.

[21] F. Dellaert, D. Fox, W. Burgard, and S. Thrun, “Monte Carlo localization for mobile robots,” in *Proceedings of the 1999 IEEE International Conference on Robotics and Automation (ICRA)*, vol. 2, 1999, pp. 1322–1328.

[22] J. Saarinen, H. Andreasson, T. Stoyanov, and A. J. Lilienthal, “Normal distributions transform Monte-Carlo localization (NDT-MCL),” in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2013, pp. 382–389.

[23] B. Ristic, M. Morelande, and A. Gunatilaka, “Information driven search for point sources of gamma radiation,” *Signal Processing*, vol. 90, no. 4, pp. 1225–1239, 2010.

[24] G. Hitz, E. Galceran, M.-È. Garneau, F. Pomerleau, and R. Siegwart, “Adaptive continuous-space informative path planning for online environmental monitoring,” *Journal of Field Robotics*, vol. 34, no. 8, pp. 1427–1449, 2017.

[25] M. Popović, T. Vidal-Calleja, G. Hitz, J. J. Chung, I. Sa, R. Siegwart, and J. Nieto, “An informative path planning framework for UAV-based terrain monitoring,” *Autonomous Robots*, vol. 44, pp. 889–911, 2020.

[26] J. Adams, A. Abrahao, L. Lagos, and D. McDaniel, “Autonomous radiation mapping using a manipulator-equipped quadruped with flexible behavior design,” *Applied Sciences*, vol. 16, no. 7, p. 3500, 2026.

[27] L. Miller, J. Keene, J. M. C. Brown, and A. Chapman, “Radioactive source seeking using Bayesian optimisation with movement penalty,” arXiv:2605.14942, 2026.

[28] E. Galceran and M. Carreras, “A survey on coverage path planning for robotics,” *Robotics and Autonomous Systems*, vol. 61, no. 12, pp. 1258–1276, 2013.

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

[30] M. Ghaffari Jadidi, J. Valls Miro, and G. Dissanayake, “Sampling-based incremental information gathering with applications to robotic exploration and environmental monitoring,” *The International Journal of Robotics Research*, vol. 38, no. 6, pp. 658–685, 2019.

[31] C. Benjamins *et al.*, “Self-adjusting weighted expected improvement for Bayesian optimization,” arXiv:2306.04262, 2023.

[32] M. Chiou, N. Hawes, and R. Stolkin, “Mixed-initiative variable autonomy for remotely operated mobile robots,” *ACM Transactions on Human-Robot Interaction*, vol. 10, no. 4, p. 37, 2021.

[33] M. Chiou, G.-T. Epsimos, G. Nikolaou, P. Pappas, G. Petousakis, S. Mühl *et al.*, “Robot-assisted nuclear disaster response: Report and insights from a field exercise,” in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2022, pp. 4545–4552.

[34] L. Methnani, M. Chiou, V. Dignum, and A. Theodorou, “Who’s in charge here? a survey on trustworthy AI in variable autonomy robotic systems,” *ACM Computing Surveys*, vol. 56, no. 7, 2024.

[35] T. Wright, A. West, M. Licata, N. Hawes, and B. Lennox, “Simulating ionising radiation in Gazebo for robotic nuclear inspection challenges,” *Robotics*, vol. 10, no. 3, p. 86, 2021.

[36] P. Proctor, C. Teuscher, A. Hecht, and M. Osiński, “Proximal policy optimization for radiation source search,” *Journal of Nuclear Engineering*, vol. 2, no. 4, pp. 368–397, 2021.

[37] H. Son, Y. Do, M. Zebrowitz, and F. Zhang, “Physics-informed radiation multi-source localization with robotic platform,” *International Journal of Intelligent Robotics and Applications*, vol. 9, no. 4, pp. 1818–1831, 2025.

[38] H. Son, K. Tan, and F. Zhang, “Physics-guided robotic radiation source localization along arbitrary measurement paths in unstructured environments,” arXiv:2606.27624, 2026.

[39] T. Lazna, P. Gabrlik, T. Jilek, and L. Zalud, “Cooperation between an unmanned aerial vehicle and an unmanned ground vehicle in highly accurate localization of gamma radiation hotspots,” *International Journal of Advanced Robotic Systems*, vol. 15, no. 1, 2018.

[40] S. Macenski, F. Martín, R. White, and J. Ginés Clavero, “The Marathon 2: A navigation system,” in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 2718–2725.

[41] B. Yamauchi, “A frontier-based approach for autonomous exploration,” in *Proceedings of the 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA)*, 1997, pp. 146–151.

Appendix A: Repository Structure and Key Components

The work described in this report lives in the `automation/` directory of the project repository. The surrounding directories are inherited or belong to other members of the team, and the boundary is stated here so that the reader can trace exactly what is claimed.

```
sensor_monitor/            inherited baseline + collaborators' work
  dds_bridge.py              robot-to-WiFi bridge (inherited; two fixes
                             from this project: auth-timeout scope,
                             client-disconnect hardening)
  sensor_bridge.py           Geiger serial-to-TCP relay (inherited)
  lidar_mapper.py            original mapper (inherited; since evolved
                             by the mapping/localisation collaborator)
  manual_bridge_protocol.py  authenticated command protocol
                             (safety/control collaborator)
  arduino/radiation_sensor/  detector firmware (inherited)

automation/                this project, except ui/
  Drift/                     drift investigation: bridge_to_ros.py,
                             slam_toolbox and KISS-ICP launch files,
                             the first radiation_mapper
  map_radiation_survey/      walk-2 tools: pose_publisher.py,
                             radiation_mapper.py, radiation_overlay.py
  intellectual_robot/        the intelligence layer (sole author):
    field_model.py             synthetic-source physics
    fake_geiger.py             wire-compatible synthetic sensor
    map_free_space.py          occupancy grid, corridor tests, focus rule
    gp_field.py                Gaussian process (Matern 3/2 + bias)
    survey_brain.py            acquisition, dwell, stopping
    path_planner.py            A* with corridor validation
    hold_walker.py             hold-to-walk motion controller
    proximity_guard.py         LiDAR freeze zone, obstacle memory
    smart_scan.py              the survey loop and operator interface
    survey_render.py           heatmap and uncertainty rendering
    tour_planner.py            multi-point tour preview
    path_trace.py              walked-route recording
    bench_drive.py             five-scenario drive backtest
    tests/                     94 automated tests in 16 files
    test/                      real-source run kit and scorer
  ui/                        operator dashboard (collaborator)
computer_control/          control-bridge test stack (collaborator)
```

Three excerpts show the core of the method as it is actually implemented. The kernel and the noise model together define the field estimator of Section 4.6:

```
def _matern32(d, ls):
    a = np.sqrt(3.0) * d / ls
    return (1.0 + a) * np.exp(-a)

def _kernel(self, A, B):
    return self.sv * _matern32(_cdist(A, B), self.ls) + self.bv

def _noise_var(self, y_usv_h, integration_s):
    counts = np.maximum(np.asarray(y_usv_h) * CPS_PER_USV_H
```

```
                        * np.asarray(integration_s), 0.5)
    return 1.0 / counts + self.nf
```

The acquisition of Section 4.7 is a single line, with the two alternatives selectable beside it:

```
if self.acquisition == "ucb":
    acq = mu_log + self.beta * boost * sigma - self.move_penalty * dist
elif self.acquisition == "pi":
    acq = _phi((mu_log - best_log - self.xi * boost) / sigma)
else:                                           # ei
    acq = ei
```

and the dwell rule of Equation 3 is implemented directly:

```
def _dwell_s(self, mu_usv_h):
    n_target = (1.0 / self.rel_err_target) ** 2      # e.g. 100 counts
    cps = max(mu_usv_h * CPS_PER_USV_H, 0.2)
    return float(np.clip(n_target / cps, 5.0, 60.0))
```

The honesty boundary of Section 4.10 can be checked from the imports alone: `gp_field.py` and `survey_brain.py` import only the calibration constant `CPS_PER_USV_H` from the physics module, never the source definitions, and no file on the estimation path opens `sources.yaml`. Only the synthetic sensor, the backtest harness and the scorer read it.

Appendix B: Run Procedures

The procedure below reproduces a scored survey from a cold start. It is written for the equipment as deployed: the robot's Jetson reachable over the local network, and the operator workstation running the project code. Machine names and addresses are those used during the campaign.

1. Jetson services (two terminals, left running throughout):

```
# terminal A -- robot bridge (pose, commands, LiDAR relay)
pkill -f dds_bridge.py
python3 ~/dds_bridge.py --interface eth0 --lidar \
        --manual-control-token-file ~/manual_token.txt

# terminal B -- detector (real-source surveys only)
pkill -f sensor_bridge.py
python3 ~/sensor_bridge.py --port /dev/ttyACM0
```

2. Pre-flight (workstation, repository root). The full test suite and the drive backtest were run before every hardware session; a session did not proceed unless both passed.

```
python3 -m pytest automation/intellectual_robot/tests/ -q   # 94 tests
python3 -m automation.intellectual_robot.bench_drive        # 5 scenarios
timeout 10 nc <jetson-ip> 9876                              # detector check
```

3. Walk 1 — map. Drive the robot slowly around the survey area, then save and quit. From this point the robot must not be power-cycled or carried: the odometry frame is anchored to power-on, and every later coordinate depends on it.

```
python3 sensor_monitor/lidar_mapper.py --bridge <jetson-ip>
```

4. Walk 2 — coarse pass. Start the pose feed and the recorder, then drive a loop covering the area of interest, including a pass through the middle. The loop defines the survey's territory: the intelligence layer will not propose measurements outside it.

```
python3 automation/map_radiation_survey/pose_publisher.py <jetson-ip>
python3 automation/map_radiation_survey/radiation_mapper.py <sensor-host> \
        --world-frame odom
```

where `<sensor-host>` is the Jetson for a real survey or the workstation itself when the synthetic source is serving. For a synthetic run the source is placed first, by code, from the map's free space, and the impostor started in its own terminal:

```
python3 -m automation.intellectual_robot.fake_geiger \
        --sources automation/intellectual_robot/sources.yaml \
        --world-frame odom --port 9876
```

5. Walk 3 — the supervised scan. The pose feed is stopped first (the command port accepts a single client) and the survey started:

```
python3 -m automation.intellectual_robot.smart_scan \
    --map sensor_monitor/data/floor_plan.png \
    --meta sensor_monitor/data/map_meta.json \
    --csv automation/map_radiation_survey/output/radiation_points.csv \
    --world-frame odom --drive --bridge-host <jetson-ip> \
    --token-file ~/manual_token.txt --publish-tf \
    --hold-timeout 1.2 --speed 0.35
```

The operator's controls during the scan: `w` approves the proposed point; holding `w` advances the robot, and releasing it stops within the hold window; `s` requests a different proposal; `q` aborts everything at any stage.

6. Ground truth and scoring. While still in the same power-on session, the robot is driven to stand beside the physical source and its reported position read from the pose feed; that position is written to `ground_truth.yaml`. The scorer then reports the distance from the survey's hotspot estimate to the recorded truth:

```
python3 automation/intellectual_robot/test/score_run.py \
    --map ... --meta ... --csv ... \
    --truth automation/intellectual_robot/test/ground_truth.yaml
```

Recording the truth with the robot itself, rather than a tape measure, places truth and readings in the same coordinate frame by construction, which is what makes the error metric meaningful under odometry drift (Section 5.4).

Appendix C: Mathematical Detail

This appendix records the derivations behind the expressions quoted in Chapter 4.

C.1 The heteroscedastic noise model. A Geiger–Müller measurement over integration time t at true rate λ counts $k \sim \text{Poisson}(\lambda t)$ events, with $\mathbb{E}[k] = \text{Var}[k] = \lambda t$. The estimator works in the logarithm of the dose rate, and the variance of $\log k$ follows from the delta method: for a function g of a random

variable with mean m, $\mathrm{Var}[g(k)] \approx g'(m)^2\,\mathrm{Var}[k]$, so with $g = \log$,

$$\mathrm{Var}[\log k] \;\approx\; \frac{\mathrm{Var}[k]}{\mathbb{E}[k]^2} \;=\; \frac{\lambda t}{(\lambda t)^2} \;=\; \frac{1}{N}, \tag{4}$$

where $N = \lambda t$ is the expected count. Each observation is therefore given noise variance $1/N_i + \sigma_0^2$, with N_i reconstructed from the reading through the calibration constant and the integration window, floored at half a count to keep the variance finite, and $\sigma_0^2 = 0.05$ a fixed floor absorbing calibration and pose error. The practical consequence is that background readings, carrying a handful of counts, are automatically trusted less than readings near a source.

C.2 The posterior. With training inputs X, log-readings $\mathbf{y}$, kernel matrix K (Equation 1) and noise matrix $\Sigma = \mathrm{diag}(1/N_i + \sigma_0^2)$, the predictive distribution at query points X_* is the standard Gaussian-process conditional

$$\mu_* = m + K_*^\top (K + \Sigma)^{-1}(\mathbf{y} - m), \tag{5}$$

$$\sigma_*^2 = k_{**} - K_*^\top (K + \Sigma)^{-1} K_*, \tag{6}$$

computed by Cholesky decomposition. The implementation was cross-checked against an independent direct-inverse implementation to an agreement of 10^{-6}. Predictions are exponentiated for display, which guarantees a non-negative dose-rate field.

C.3 Expected improvement and the stopping metric. With incumbent best log-value y^+ and margin ξ, writing $z = (\mu_* - y^+ - \xi)/\sigma_*$, expected improvement has the closed form

$$\mathrm{EI}(\mathbf{x}) = \sigma_*\big(z\,\Phi(z) + \phi(z)\big), \tag{7}$$

with Φ and ϕ the standard normal distribution and density. The stopping metric normalises the largest expected improvement over reachable candidates by the observed dynamic range of the data, $\max \mathrm{EI}/(\max \mathbf{y} - \min \mathbf{y})$, and the survey ends when this fraction stays below 0.02 for three consecutive proposals. Because $\mathrm{EI} \to 0$ as $\sigma_* \to 0$, saturation of the posterior variance triggers the same criterion.

C.4 The dwell rule. The relative standard error of a Poisson count of N events is $\sqrt{N}/N = 1/\sqrt{N}$, so a target relative error e requires $N \geq 1/e^2$; the default $e = 0.10$ gives $N = 100$ counts. The predicted local rate converts the count target to a duration, clamped to 5–60 s. At the weak sources surveyed here the rule sat at its upper clamp, which is itself informative: the statistics, not impatience, set the measurement time.

C.5 The synthetic field. The impostor sensor computes, from the robot's live position, the rate

$$\lambda(\mathbf{x}) = b + \sum_j \frac{s_j}{d_j^2 + h^2}, \tag{8}$$

with b the configured background, s_j source strengths, d_j horizontal distances and $h = 0.30$ m the detector height; a Poisson draw at the detector's calibration and integration converts the rate to the quantised readings a real tube produces. The same expression with A and b free is the curve fitted to the real-source data in Section 5.5, where it accounts for $R^2 = 0.742$ (run B) and 0.391 (run C) of the variance.

Appendix D: Full Experimental Data

The tables below record the campaign's complete measured record, retrievable from the archived session data.

Table 8 The three scored surveys: recording statistics.

	Rehearsal (A)	Real, first (B)	Real, final (C)
Date (2026)	3 Aug	3 Aug	4 Aug
Readings	300	450	495
Duration (min)	5.0	7.5	8.2
Dose min (μSv/h)	0.553	0.038	0.028
Dose mean	5.99	0.064	0.059
Dose max	26.89	0.111	0.091
95th percentile	25.78	0.104	0.086
Map extent (m)	16.5×16.6	5.1×11.3	4.8×9.4
Truth (x, y)	(-0.97, -0.81)	(22.75, 16.67)	(22.90, 15.97)
Estimate (x, y)	(-0.72, -0.66)	(22.97, 16.39)	(22.87, 15.86)
Error (m)	**0.29**	**0.36**	**0.12**

Table 9 Coverage statistics for the two real-source surveys.

	Run B	Run C
Convex hull of readings (m^2)	1.25	2.34
Fraction of mapped free space	2.3%	5.7%
Mean nearest-neighbour spacing (m)	0.008	0.011
Nearest reading to truth (m)	0.004	0.001
Readings within 0.5 m of truth	178	225
Readings within 1.0 m of truth	248	244
Walk-3 distance walked (m)	—	2.18

Table 10 Model validation: leave-one-out prediction and calibration.

	Run B	Run C
LOO RMSE (μSv/h)	0.0084	0.0100
Relative RMSE	13.2%	17.0%
Coverage at $\pm1\sigma$, posterior only	62.7%	53.3%
Coverage at $\pm1\sigma$, incl. noise	100%	100%

Finally, the behaviour of the source-marking audit (Section 5.10) on the two frozen datasets: on the synthetic background the deployed ratio gate passed (max μ/median $\mu = 4.88$) and published three spurious peaks; on the final real survey it failed by 3×10^{-4} of the threshold (1.4996 against 1.5) and published none. The significance rule chosen to replace it ($z \geq 4$, working interval approximately 3.4–4.7) yields zero peaks on the background, one peak 0.063 m from truth on run C, and one peak 0.141 m from truth on run B.

Table 11 Posterior uncertainty at the true source position against number of readings (chronological prefixes). Saturation by roughly 300 readings in both runs.

n	Run B $\sigma_{\log}$	Run C $\sigma_{\log}$
25	1.118	1.144
50	0.697	1.128
100	0.308	0.305
200	0.071	0.117
300	0.057	0.066
all	0.057	0.066

Table 12 Kernel length-scale sensitivity: hotspot error (m) when the frozen data is refitted at each length scale.

	$\ell = 0.5$ m	$\ell = 1.0$ m (used)	$\ell = 2.0$ m
Run B	0.357	0.357	0.827
Run C	2.061	0.117	0.117

Table 13 Offline acquisition comparison on the final run's frozen data. The hotspot estimate is identical by construction; the first proposals differ. Later proposals were enumerated by masking each accepted pick, which also escalates the exploration weight, and are therefore indicative rather than a replay of live behaviour.

Method	Error (m)	First three proposals	Propose time (ms)
UCB	0.117	(22.8, 11.7), (22.3, 12.2), (22.3, 15.2)	338
PI	0.117	(22.3, 12.2), (22.8, 11.7), (22.3, 15.2)	317
EI	0.117	(22.8, 11.7), (22.3, 12.2), (22.3, 15.2)	325

Table 14 Drive backtest, five scenarios, as re-executed on the final code. All pass.

Scenario	Outcome
Clear corridor	Arrived, 0.14 m from target
Mapped obstacle	Arrived, 5.1 m detour against 2.0 m straight line
Person steps in	Froze 6.8 s for the 5 s blockage (obstacle memory), resumed, arrived
Full survey	Six checkpoints, 134 readings, hotspot error 0.25 m
Protocol loopback	Arrived within 0.13 m; stop command confirmed delivered