MANCHESTER 1824
The University of Manchester

Corroborative Verification and Validation of an Autonomous Warehouse Inspection Robot Using High-Fidelity Simulation and Probabilistic Model Checking

A dissertation submitted to The University of Manchester for the degree of

Master of Science in Robotics

in the Faculty of Science and Engineering

Year of submission

2025

Student ID

10818421

School of Engineering

Contents

Word count: 6361

Abstract

Assuring the reliability, safety, and effectiveness of autonomous robots demands Verification and Validation (V&V) across multiple abstraction levels. Corroborative V&V combines heterogeneous strands - formal modelling, simulation, and physical testing - so that the strengths of one counterbalance the limitations of another. However, widely used low-fidelity simulators often miss perception artefacts and contact-rich dynamics, weakening confidence that reported behaviours will transfer to hardware. This dissertation implements a two-strand corroborative V&V pipeline for a warehouse-inspection unmanned ground vehicle tasked with visiting six waypoints in strict order, detecting fiducial markers, and returning to the dock under battery constraints. A discrete-time Markov chain is paired with a high-fidelity NVIDIA Isaac Sim implementation and seven assurance metrics are evaluated. Each metric is computed independently within each strand - probabilistically via model checking in PRISM, and from 276 simulation runs with Jeffreys intervals - and then compared for corroboration. The results show that all seven V&V metrics corroborate the assurance case. Residual discrepancies are primarily attributed to battery-model abstractions and edge-case docking arbitration, while conditional perception success saturates near unity in the evaluated setting. A qualitative assessment indicates that Isaac Sim's photorealistic sensors and contact dynamics strengthen the perception-in-the-loop and energy-related evidence relative to lower-fidelity simulators, albeit at higher computational cost.

Declaration of Originality

I hereby confirm that no portion of the work referred to in the thesis has been submitted in support of an application for another degree or qualification of this or any other university or other institute of learning.

Copyright Statement

i The author of this thesis (including any appendices and/or schedules to this thesis) owns certain copyright or related rights in it (the "Copyright") and s/he has given The University of Manchester certain rights to use such Copyright, including for administrative purposes.

ii Copies of this thesis, either in full or in extracts and whether in hard or electronic copy, may be made *only* in accordance with the Copyright, Designs and Patents Act 1988 (as amended) and regulations issued under it or, where appropriate, in accordance with licensing agreements which the University has from time to time. This page must form part of any such copies made.

iii The ownership of certain Copyright, patents, designs, trademarks and other intellectual property (the "Intellectual Property") and any reproductions of copyright works in the thesis, for example graphs and tables ("Reproductions"), which may be described in this thesis, may not be owned by the author and may be owned by third parties. Such Intellectual Property and Reproductions cannot and must not be made available for use without the prior written permission of the owner(s) of the relevant Intellectual Property and/or Reproductions.

iv Further information on the conditions under which disclosure, publication and commercialisation of this thesis, the Copyright and any Intellectual Property and/or Reproductions described in it may take place is available in the University IP Policy (see `http://documents.manchester.ac.uk/DocuInfo.aspx?DocID=24420`), in any relevant Thesis restriction declarations deposited in the University Library, The University Library's regulations (see `http://www.library.manchester.ac.uk/about/regulations/`) and in The University's policy on Presentation of Theses.

Acknowledgements

I am deeply grateful to my dissertation supervisor, Professor Watson, for guidance and thoughtful feedback throughout this work. I also thank Dr B. Abeywickrama for sharing their expertise and answering my questions, and I gratefully acknowledge the support of Research IT - specifically Nitesh Deshmukh - at The University of Manchester.

1 Introduction

1.1 Background and Motivation

Autonomous robots are increasingly deployed in industrial settings, in particular in logistics and warehousing, where reliability, safety, and effectiveness are non-negotiable [1], [2]. Assuring these behaviours for an autonomous system requires Verification and Validation (V&V). Verification ensures the alignment of a system with respect to its formal specifications, and validation ensures the system meets stakeholder requirements in its intended context of use.

Several V&V techniques exist - namely formal modelling, simulation, and physical testing - but using any one in isolation necessitates a compromise between examining the full state space of a system and modelling the system in satisfactory detail. To address this, engineers often rely on several fragmented V&V techniques, resulting in uncertainty about whether the system is truly reliable, safe, and effective. Webster *et al.* [3] therefore propose a corroborative approach to V&V (CV&V): combine these complementary heterogeneous techniques, cross-check their results, and iteratively refine models so that the strengths of one technique compensate for the limitations of another, strengthening the overall assurance argument.

High-fidelity simulators now promise to decrease the level of abstraction required for the simulation leg of CV&V by implementing increased photorealism, physics and sensor fidelity, determinism, and GPU-accelerated scalable scenario generation compared with legacy simulators. A higher-fidelity simulator could serve to address the recognised industrial and academic need for rigorous and tool-agnostic V&V of increasingly complex systems [2]. Yet many CV&V pipelines still rely on lower-fidelity simulators [3]–[7]. This work shows how a high-fidelity simulator, NVIDIA Isaac Sim [8], can be used in the CV&V pipeline, and qualitatively evaluates it against commonly used lower-fidelity tools.

1.2 Aims and Objectives

This dissertation aims to complete CV&V of an autonomous warehouse inspection unmanned ground vehicle (UGV) operating in a cluttered warehouse, in order to demonstrate the utility of high-fidelity simulators - specifically NVIDIA Isaac Sim. The UGV must visit six waypoints in strict order while avoiding obstacles. At each waypoint it attempts to detect a fiducial marker on a box using on-board vision and, on success, the observation is logged to an external server. The mission ends when all waypoints are visited, or earlier if the state of charge (SoC) drops below a predefined threshold, in which case the UGV aborts the patrol and returns to its dock. Assurance-oriented metrics are defined - including the probability of visiting all waypoints in order, detecting all fiducial markers, returning to the dock, and docking due to low SoC. The CV&V study comprises a high-fidelity Isaac

Sim implementation for the simulation strand and a discrete-time Markov chain (DTMC) for the formal-model strand, with the metrics evaluated independently on each strand and compared to corroborate the assurance case.

While CV&V commonly draws on three strands of evidence (formal model, simulation, and physical experimentation), this dissertation intentionally focuses on the first two as physical experimentation is out of scope. However, the framework remains readily extensible: metrics are aligned across all strands, and the data analysis pipeline accepts new evidence sources as drop-in providers. As a result, experimental logs from future test-rig or field trials can be ingested with minimal changes to the analysis workflow.

In summary, the main contributions of this work are:

1. A high-fidelity NVIDIA Isaac Sim implementation and a probabilistic DTMC formal model of an autonomous warehouse inspection UGV operating in a cluttered warehouse.
2. An implementation of CV&V that quantitatively compares assurance metrics across the simulation and formal model and explains discrepancies.
3. An exploration of the use of Isaac Sim to advance the field of autonomous system CV&V.

1.3 Report Structure

The remainder of this dissertation is organised as follows. Section 2 provides an in-depth review of the literature on CV&V and the use of high-fidelity simulation in CV&V. Section 3 details the warehouse inspection case study, the Isaac Sim setup, the DTMC formulation, and the definition of CV&V metrics. Section 4 presents and discusses results, comparing metrics across strands and evaluating Isaac Sim. Finally, Section 5 concludes the dissertation and outlines avenues for future work.

2 Literature Review

2.1 Corroborative Verification and Validation

V&V evidence for autonomous systems is typically assembled across three abstraction levels - formal models, simulation, and test-rig experiments - each with distinct strengths and limitations. Formal models provide logically rigorous results by exhaustively analysing an abstract state space, but are constrained by modelling assumptions and state-space explosion [9]. Simulation offers tunable fidelity and broad, repeatable scenario coverage at comparatively low physical risk and cost, but retains residual modelling error. Test-rig experiments expose actual hardware under controlled

conditions, yielding evidence that most closely mirrors operational realities but at greater expense, slower iteration, and reduced repeatability.

CV&V addresses these trade-offs by treating the heterogeneous V&V techniques as complementary strands of evidence whose results are cross-checked and iteratively reconciled. Webster *et al.* [3] demonstrate how model checking, simulation-based testing, and user studies can expose inconsistent findings, which then trigger targeted refinement of requirements and models until the strands converge to a defensible tolerance.

Other work explores variations of corroborative approaches by combining formal analysis with simulation across several domains. In unmanned aerial vehicle certification, Webster *et al.* [10] pair model checking of the autonomous logic with flight simulation. They verify operational rules, use counterexamples to revise the controller, and show how simulation evidence can support certification. They also recognise limitations, including simplified environment models, a proof-of-concept scope, and the need for tool qualification before regulators accept the evidence. For the Curiosity Mars rover, Cardoso *et al.* [7] build a heterogeneous stack that includes formal analysis of the agent that makes decisions, proof obligations over the software interfaces between agent and simulator, checks of the messaging protocols, and runtime monitors inside a physics-based simulator. This exemplifies how different strands can catch different classes of defects, but it leaves out power and environmental hazards, and stresses that runtime monitoring is necessarily incomplete. In automated driving, Schwammberger *et al.* [6] integrate formal reasoning about traffic rules with assertion checking over simulated drives, using proofs to target simulations at boundary cases. Abeywickrama *et al.* [4] extend CV&V to robot swarms with a top-down–bottom-up loop, in which temporal-logic properties are checked on population-level models, parameters are fitted from large simulation batches, and spot checks are performed on real robots. In inspection robotics, the approach has been implemented for a nuclear UGV by pairing goal structured notation assurance patterns (strict-order patrol and global avoidance) with cross-checks between a probabilistic formal model and a simulator [5]. Together, these studies illustrate a common pattern: use formal analysis to focus simulation and experiments, use empirical evidence to calibrate and challenge models, and iterate until the evidence strands support a coherent assurance case.

2.2 High-Fidelity Simulation for CV&V

Prior CV&V studies have relied on lower-fidelity legacy simulators for their simulation V&V evidence [3]–[5], [7]. Gazebo [11] became a de facto standard because it is open-source and native to the Robot Operating System (ROS), consumes Unified Robot Description Format (URDF) and Simulation Description Format (SDF) models directly, supports a rich plugin ecosystem, and offers multiple

CPU physics back ends. In Gazebo Classic, these back ends include ODE, Bullet, and Simbody, and DART is used in Gazebo. This ecosystem enables reproducible, headless experimentation on modest hardware and remains effective for numerous planning and control tasks. However, lower-fidelity pipelines can struggle to reproduce perception artefacts and contact-rich phenomena that strongly influence end-to-end behaviour. Simplified contact, friction, and restitution models can bias energy costs, stability, and recovery dynamics, and cross-engine comparisons report large variance and non-conservative errors even on simple benchmarks [12]–[14]. On the perception side, limited photorealism, lighting realism, sensor noise, and lens effects can yield vision-performance gaps relative to higher-fidelity alternatives, thereby reinforcing a persistent simulation–reality domain gap [15].

A newer generation of high-fidelity simulators emphasises physically based rendering, ray-traced sensors, richer materials, and modern rigid-body solvers - prominently CARLA [16] for driving and AirSim [17] for aerial/ground vehicles - aimed at testing closed-loop autonomy under realistic sensing and dynamics. In this work, NVIDIA Isaac Sim [8], a reference application built on NVIDIA Omniverse, is adopted as a higher-fidelity alternative for CV&V. Its GPU-accelerated sensor simulation uses hardware ray tracing (Omniverse RTX), enabling cameras and lidars to interact with scene materials and lighting in a physically based manner (such as reflections, refractions, and motion effects), better exposing perception-driven failure modes. On the dynamics side, the underlying NVIDIA PhysX 5 engine [18] provides GPU-accelerated rigid bodies, a Temporal Gauss–Seidel solver for stable contacts at larger time steps, and reduced-coordinate articulations that tightly enforce joint constraints. For CV&V, these features are attractive because they yield perception streams and contact interactions closer to those observed on hardware, support deterministic stepping with fixed seeds for repeatable experiments, and scale scenario sweeps on GPUs [12].

Despite its advantages, high-fidelity simulation entails trade-offs. Achieving photorealistic sensing and high-fidelity dynamics requires capable GPUs [19] and incurs higher iteration costs than lower-overhead toolchains, potentially constraining state-space exploration [3]. GPU-accelerated rendering can weaken strict determinism unless seeds and scheduling are carefully controlled, and the USD/Omniverse workflow has a steeper learning curve than SDF-centric pipelines. Finally, while higher fidelity can narrow perception gaps [15], it does not remove them: camera response curves, sensor readout timing, and scene-specific materials often require calibration when absolute accuracy is needed [20]. In sum, high-fidelity simulation complements formal analysis and test-rig evidence within CV&V by improving the realism of perception and contact dynamics, provided that its computational and workflow costs are managed.

3 Methods

As in Abeywickrama *et al.* [5], two goal structuring notation-based assurance patterns are implemented: *strict order patrol* and *global avoidance* [21], [22]. Section 3.1 applies these patterns to a warehouse inspection UGV case study, realised both as a high-fidelity Isaac Sim implementation (Section 3.2), and a DTMC formal model (Section 3.3). Section 3.4 then defines the independently measurable assurance metrics used for CV&V.

3.1 Warehouse Inspection Mission

The system under test is a Nova Carter P1 UGV [23] operating in a realistic warehouse adapted from Isaac Sim's `full_warehouse` Universal Scene Descriptor (USD) asset. The warehouse measures 54×32 m and comprises an open southern staging area with scattered pallets and a parked forklift, and a northern storage area of seven 16 m north–south rack runs forming six aisles. Two aisles contain mobile ladder units (one with a nearby bystander), one aisle is partially blocked by a forklift, and another is fully blocked by a debris field of fallen boxes. The Carter's docking station is centred on the eastern wall. Rendered views of the environment and the occupancy map sampled at the Carter's lidar height ($z = 0.62$ m) are provided in Appendix A.

The software stack is built on ROS 2 Humble [24] and adapted from the `navigation/carter_navigation` package provided in the Isaac Sim ROS 2 Humble workspace [8]. Missions start with the robot fully charged and docked. The UGV patrols six waypoints in strict numerical order while avoiding obstacles. If a collision is detected or the navigation stack reports a failure, the behaviour tree `navigate_to_pose_w_replanning_and_recovery.xml` is invoked to execute standard Nav2 clearing and recovery actions such as `Spin`, `BackUp`, and `Wait` [25]. At each waypoint, it processes ten consecutive frames from the front Hawk stereo camera to detect a fiducial marker placed on a visible crate or box. As shown in Appendix A, markers are 0.25 m square AprilTags from the tag36h11 family [26], each with a unique ID matching the waypoint number. The detection node is adapted from the CUDA-accelerated AprilTag detection and pose estimation package, `isaac_ros_apriltag` [27]. Upon correct detection, the ID is transmitted and logged on an external server. After visiting the sixth waypoint - or earlier if the SoC drops below 20% - the robot terminates the patrol and returns to the dock.

The Carter is equipped with the full Nova Orin sensor suite: four side-mounted Hawk stereo cameras, four side-mounted Owl fisheye cameras, on-board IMUs, front and rear RPLidar S2E 2D lidars, and a top-mounted Hesai XT32 3D lidar. The platform is powered by a Jetson AGX Orin, measures $722 \times 500 \times 556$ mm and has a mass of 51.2 kg. Its maximum speed is $3.3\ \mathrm{m\,s^{-1}}$ and the nominal battery capacity is 1033 Wh, enabling over 8 hours of operation. For experimental throughput during

data collection, the battery capacity is scaled to 6.75 Wh, yielding at least three minutes of runtime, sufficient for short inspection missions under nominal conditions.

3.2 High-Fidelity Simulation

The simulation is implemented in NVIDIA Isaac Sim 4.5.0 [8], which provides high-fidelity physics and sensor models (see Section 2.2). The case study is ported to simulation with an explicit battery model implemented as a ROS 2 node [28]. The instantaneous electrical power $P(t)$ is modelled as

$$P(t) = P_{\text{idle}} + k_v \left\|\mathbf{v}(t)\right\| + k_\omega \left|\omega(t)\right|, \tag{1}$$

where P_{idle} is the idle power draw (W), $\|\mathbf{v}(t)\|$ is planar chassis speed ($\mathrm{m\,s^{-1}}$), $|\omega(t)|$ is the yaw rate ($\mathrm{rad\,s^{-1}}$) and k_v and k_ω are the linear and angular power coefficients respectively. P_{idle} is estimated from charge-time data: with a 210 W charger, a full charge of 1033 Wh takes $T{=}9$ hours while idling (compared to 5 hours in sleep). Assuming negligible other losses, the implied idle load is

$$P_{\text{idle}} \approx 210 - \frac{1033}{T} = 210 - 115 \approx 95 \text{ W}. \tag{2}$$

The linear and angular power coefficients, k_v and k_ω, are calculated from first-principles estimates (mass, wheelbase width, rubber–concrete friction, drivetrain efficiency), yielding

$$k_v = 30 \text{ W } (\mathrm{m\,s^{-1}})^{-1}, \qquad k_\omega = 15 \text{ W } (\mathrm{rad\,s^{-1}})^{-1}. \tag{3}$$

These values are consistent with the 130 W literature value for Carter's cruise load [23].

A lightweight logger node subscribes to the necessary ROS 2 topics and watches for relevant events, namely run start/end, waypoint arrivals (`waypoint,`i logs arrival at waypoint i), detected AprilTag IDs (`tag,`i logs successful detection of the AprilTag at waypoint i), recovery enter/exit, and docking start/completion. The node then writes these events, together with timestamps and current SoC values, to an external CSV file. Runs terminate on docking or SoC depletion. After each run, the navigation stack is reset, SoC is set to full, and AprilTag ID memory is cleared. To maximise experimental throughput, the static Isaac Sim stage is not reloaded between runs.

The simulations were executed on an AWS NVIDIA Isaac Sim Development Workstation AMI accessed via the NICE DCV client. The instance ran Ubuntu 22.04 using an AMD EPYC 7R13 server-class CPU and an NVIDIA L40S GPU (48 GB, CUDA 12.4). The ROS 2 workspace is available at `https://github.com/EuanBaldwin/Isaac_sim_ws`.

3.3 DTMC Formal Model

3.3.1 Model Logic

The strict-order patrol is formalised as a DTMC in PRISM, a probabilistic model checker [29]. The model abstracts one patrol episode over the ordered route $0{\to}1{\to}2{\to}3{\to}4{\to}5{\to}6{\to}0\,(\text{dock})$ with stochastic navigation recoveries, tag detection at each waypoint, and battery-aware early docking. The model is available at `https://github.com/EuanBaldwin/prism-carter`.

The state comprises: a phase flag (`phase` $\in 0,1,2$ for patrol, docked, failure), the current waypoint (`wp` $\in 0,\ldots,6$), a discretised state-of-charge (`bat` $\in 0,\ldots,$ `MAX_BAT`; 1 unit$=0.1\%$ SoC), a time step counter $t \in 0,\ldots,$ `MAX_TIME`, latches for `recovery`, `scanned`, `already_docking`, `set_bat_low`, and six booleans `seen1..6` capturing per-waypoint detection outcomes.

Movement along edge $i{\to}j$ is represented by guarded commands, `[move_ij]`. Conditional on eligibility (no active recovery; within the time horizon; sufficient SoC), a Bernoulli "collision" trigger with probability $p_{\text{recovery},ij}$ either sets the `recovery` latch or succeeds, debiting a segment-specific energy cost `cost_ij` and advancing `wp`. A single scan attempt per waypoint k, `[scan_k]`, succeeds with probability p_{detect} and latches `seen`k; scans are only permitted while above the low-SoC threshold and must be completed before leaving a waypoint.

When `recovery` is `true`, a recovery action, `[recovery]`, consumes `cost_recovery` and succeeds with probability $p_{\text{recovery\_success}}$; failure transitions to the absorbing failure state, `phase=2`. This mechanism realises the *global avoidance* assurance pattern as a stochastic hazard with a bounded, explicitly modelled recovery subroutine.

Energy costs are debited discretely on successful moves and on each recovery attempt. To reflect an operational reserve policy, patrol edges $i{\to}j$ are guarded by `docking_soc` which only enables the movement step if, after paying `cost_ij`, the battery would remain $\geq$ `LOW_BAT`. To align with the mission specifications, `LOW_BAT` is set to 20%. If the reserve SoC were to be violated, the SoC is clamped to `LOW_BAT` and the early-docking path is taken. The latch `already_docking` ensures that if a recovery is triggered during the docking leg, then the model resumes docking rather than re-evaluating the patrol policy. Failure guards trigger a transition to failure on timeouts ($t \geq$ `MAX_TIME`) or when an obligatory action cannot be afforded (insufficient SoC to attempt a required recovery or to perform the docking leg). Two absorbing modules, `phase=1` for docking success, and `phase=2` for failure, close the Markov chain.

Additionally, several convenience property labels are specified for readability. These include `all_tags`, the conjunction of all six AprilTags being detected, and `all_wp`, the conjunction of all

six waypoints being visited.

3.3.2 Model Parameterisation

The model is driven by several independently derived parameters that specify the probability of AprilTag detection, the per-leg probability of recovery, the probability of recovery success, and the per-leg SoC cost of movement.

The detection success $p_{\text{detect}}{=}0.99988$ represents the probability that ten independent AprilTag frame checks at a waypoint yield a correct detection. This is based on the tag36h11 recall statistics reported by Innocenti *et al.* [30], and corroborated by Wang *et al.* [26].

The per-leg probability of recovery initiation, $p_{\text{recovery},ij}$, is modelled as a memoryless Poisson process per metre travelled:

$$p_{\text{recovery},ij} = 1 - e^{-\lambda L_{ij}}, \tag{4}$$

where λ is the literature value for the probability of recovery per metre calculated to be 0.00279 from the Nav2 Marathon dataset [31], and L_{ij} is the approximate length of the leg while avoiding obstacles. The likelihood of recovery success, $p_{\text{recovery\_success}}$, is also estimated from the Nav2 Marathon dataset; however, because the raw estimated value of recovery success is unity over the 168 recoveries, the lower limit of the equal-tailed 95% Jeffreys interval (Beta$(\frac{1}{2}, \frac{1}{2})$) [32] is taken as a conservative estimate. This results in a $p_{\text{recovery\_success}}$ of 0.9852.

The per-leg and recovery SoC costs are derived from an abstracted version of the same model that drives the simulation's power use (Equation (1)). Per-leg costs are calculated assuming the Carter moves at a mean speed of 1.0 m s^{-1} and spends an additional 5 s idling as it arrives and leaves waypoints. This 5 s idle time also helps account for the slowdown of the robot near waypoints and the rotation required to reach the correct pose. The recovery SoC cost is calculated based on the robot's Nav2 recovery behaviour tree [25] as a 5 s `Wait`, a 0.3 m `BackUp` at 0.15 m s^{-1}, and a 1.57 rad `Spin` at 0.8 rad s^{-1}.

3.4 Metrics for CV&V

As shown in Table 1, seven assurance case metrics are defined to quantitatively compare the DTMC formal model and the high-fidelity simulation. The translation of these metrics to hypothetical physical experiments is also outlined, although such experiments are not undertaken in this work. Together these assurance metrics quantify navigation and inspection robustness, mission success (defined as all AprilTags being detected and the Carter successfully returning to the dock), and SoC management.

Table 1. Definition of the metrics across the formal model, simulation, and physical experiments.

Metric	Formal model (PCTL reachability)	Simulation (logged events)	Physical experiments (empirical events)
Returned to dock	$P{=}?\big[F\,(\texttt{phase=1})\big]$	`waypoint,0`	returned to the dock
All waypoints	$P{=}?\big[F\,\texttt{all\_wp}\big]$	`waypoint,1..6` in strict order	visited all waypoints in strict order
All tags $\mid$ all waypoints	$\frac{P{=}?\big[F\,(\texttt{all\_tags}\wedge\texttt{all\_wp})\big]}{P{=}?\big[F\,\texttt{all\_wp}\big]}$	`tag,1..6` $\mid$ `waypoint,1..6` in strict order	detected all six tags given all waypoints visited in strict order
Mission success	$P{=}?\big[F\,(\texttt{docked}\wedge\texttt{all\_tags})\big]$	`waypoint,0` $\wedge$ `tag,1..6`	docked and detected all six tags
Docking low SoC	$P{=}?\big[F\,(\texttt{docking\_soc}\ \wedge \neg\texttt{already\_docking})\big]$	`dock,low_soc`	triggered early docking due to SoC $\leq 20\%$
Terminal low SoC	$P{=}?\big[F\,(\texttt{bat}\leq\texttt{LOW\_BAT})\big]$	`run_ended` with `soc` $\leq 20\%$	mission terminated with SoC $\leq 20\%$
Recovery	$P{=}?\big[F\,\texttt{recovery}\big]$	at least one `recovery`	at least one navigation recovery required

For DTMCs, PRISM's property language reduces to Probabilistic Computation Tree Logic (PCTL). Each metric can then be represented as a PCTL reachability query using the "eventually" operator F, where $P{=}?\,[\,F\,\phi\,]$ returns the probability that a state satisfying ϕ is eventually reached. The conditional perception metric is expressed as a ratio of two reachabilities to isolate perception given successful navigation.

Simulation metrics are described as logged events and a post-processing script is run over the simulation's CSV data logs to measure the metrics. The script also returns the simulation's confidence interval using the equal-tailed 95% Jeffreys interval (Beta$(\frac{1}{2},\frac{1}{2})$ prior) as well as several lower-level metrics that could help with aligning the models. The Jeffreys interval is used as it maintains good coverage for small sample sizes and extreme proportions [32].

4 Results and Discussion

4.1 Corroborative Verification and Validation

Table 2 summarises the seven CV&V assurance metrics defined in Section 3.4, reported both as PCTL reachabilities from the DTMC and as estimates from 276 simulation runs. Corroboration is accepted when the formal PCTL reachability lies within the equal-tailed 95% Jeffreys interval of the simulation estimate [3]. Under this criterion, all metrics corroborate across the formal model and the simulation. These results therefore provide corroborative verification and validation between the DTMC formal model and the high-fidelity simulation for the mission and environment considered.

Table 2. Formal and simulated probabilities for the defined metrics and whether they corroborate. LL/UL = lower/upper limit of the Jeffreys interval.

Metric	Formal model PCTL reachability	Simulation n	Estimate	LL	UL	Corroborates
Returned to dock	0.994	276	0.996	0.983	1	True
All waypoints	0.961	276	0.938	0.905	0.962	True
All tags | all waypoints	0.999	259	1	0.990	1	True
Mission success	0.958	276	0.938	0.905	0.962	True
Docking low SoC	0.035	276	0.058	0.035	0.090	True
Terminal low SoC	1	276	1	0.991	1	True
Recovery	0.286	276	0.301	0.249	0.357	True

4.2 Analysis of Results

Although all seven metrics corroborate, several lie near the limits of their confidence intervals, and the mechanisms of agreement vary by metric. Accordingly, the analysis examines (i) failure modes and returns to dock; (ii) recoveries; (iii) the relative influence of perception versus navigation; and (iv) SoC modelling.

The simulation achieves a *returned to dock* rate of 0.996 (95% CI $[0.983, 1.000]$), corresponding to a single non-returning run out of 276. Although this corroborates the formal probability 0.994, the observed simulation failure mechanism differs from the DTMC's dominant failure mode. In the single simulation failure, the Nova Carter reached waypoint 6 at precisely the time the SoC crossed the low-SoC threshold and the simultaneous issuance of the regular docking command and the low-SoC docking command caused both commands to be rejected, leaving the robot idle at waypoint 6 until SoC exhaustion. By contrast, *returned to dock* failures in the DTMC are predominantly induced by unsuccessful recoveries. This does not undermine the corroboration result but indicates both a refinement opportunity for explicit arbitration between competing docking triggers in the formal model and the need for an additional validation guard that prioritises safety-critical low-SoC docking in the simulation.

The probability of at least one recovery is 0.301 in simulation versus 0.286 formally; this difference ($+0.015$) lies well inside the 95% CI $[0.249, 0.357]$. This pattern supports two aspects of the formalisation. First, it shows that the distance-proportional, memoryless recovery trigger (Equation (4)) serves as a strong operational approximation in a cluttered warehouse. Second, it demonstrates that the Jeffreys-regularised recovery success probability ($p_{\text{recovery\_success}}{=}0.9852$) is well calibrated relative to Nav2's behaviour tree implementation. Partial aisle blockages in the warehouse likely elevate local hazard rates above the Nav2 Marathon value used to set λ [31]. This could explain the small positive bias observed in the simulation's recovery rate. Consequently, the hazard-rate parameter should be treated as environment-dependent and calibrated using clutter statistics, with

a sensitivity analysis over λ reported to demonstrate that recovery-related corroboration is robust to plausible variation.

Both the formal model and the simulation corroborate that mission outcomes (*mission success*) are dominated by navigation rather than perception. Conditional on visiting all waypoints, *all tags* is unity in simulation (95% CI $[0.990, 1.000]$), with a formal value of 0.999. In contrast, *all waypoints* is 0.938 in simulation (95% CI $[0.905, 0.962]$) versus 0.961 formally, and *mission success* tracks *all waypoints* accordingly (both 0.938 in simulation; formal 0.958). Given the large (25 cm) well-lit tags viewed near normal incidence - with ten frames per waypoint via the Isaac ROS CUDA pipeline [27] - missed detections are not observed in this setup. For CV&V, this implies that, in the evaluated environment, the assurance case is placing greater stress on navigation and resource management than on perception; accordingly, assurance arguments and test budgets should prioritise waypoint reachability and SoC margins, with targeted perception stress-tests reserved for degraded visual conditions.

Table 3. Formal model and mean (M) simulated fractional SoC movement costs. $\Delta =$ formal model $-$ M.

Cost term	Formal model	Simulation				Δ
		n	M	SD	SE	
`cost_06`	0.742	259	0.761	0.007	0.000	-0.019
`cost_dock`	0.153	275	0.146	0.011	0.001	0.007
`cost_recovery`	0.036	98	0.014	0.016	0.002	0.022
`cost_total`	0.907	276	0.906	0.024	0.001	0.001

The difference in *all waypoints* (-0.023 simulation-formal) is mirrored by a $+0.023$ simulation–formal difference in *docking low SoC* (a two-thirds relative increase over the formal 0.035). To probe this, SoC cost components were compared (Table 3). The per-leg cost to reach waypoint 6 excluding recoveries (`cost_06`) is higher in simulation than in the formal model (difference -0.019), indicating a modest underestimation of per-segment SoC in the DTMC. Two explanations are plausible. First, turning and approach overheads near waypoints may exceed the 5 s idle allowance used to derive DTMC leg costs. Second, path curvature and small oscillations from local planning and obstacle inflation increase both the distance travelled and the yaw actuation (the k_v and k_ω terms in Equation (1)) relative to straight-line estimates.

The near-equality in `cost_total` arises because the DTMC's lower movement cost is offset by higher, fixed docking and recovery costs. However, the simulation exhibits substantial variability in recovery costs (the standard deviation is comparable to the mean), whereas the formal model applies a constant penalty per recovery. This highlights a modelling limitation: although the fixed recovery cost suffices for top-level corroboration, it cannot reproduce the observed per-run variability in SoC. A more robust abstraction would model recovery costs as a distribution - for example, state-dependent

rewards conditioned on local clutter or recovery duration - and include explicit turning and settling overheads in per-leg costs.

4.3 Evaluation of NVIDIA Isaac Sim

Use of NVIDIA Isaac Sim provided capabilities that shaped the study design and strengthened the credibility of the resulting CV&V evidence. Compared with legacy rasterisation-based simulators such as Gazebo, Isaac Sim combines high-fidelity dynamics (PhysX-based contact and friction, articulated rigid-body constraints) with photorealistic, RTX-accelerated sensor simulation. This enabled perception-in-the-loop evaluation, in which the robot's exact camera pipeline (intrinsics, rolling-shutter timing, exposure, distortion, noise, and motion blur) was exercised under physically plausible illumination and occlusion. This capability was essential for arguing that the perception components of the assurance case are grounded in the reality that the physical system will encounter, rather than in idealised renderings.

The primary empirical effect in the setup is visible in the conditional perception metric *all tags* $\mid$ *all waypoints*, which approached unity in both the DTMC and the simulation (Table 2). As the tags were fairly large, well-lit, and viewed at near normal incidence with ten frames per waypoint, the task sits in an "easy" regime for AprilTags [26]. In this sense, photorealism was not required to reach saturation in the reported metric. However, the photorealism of Isaac Sim remained consequential. First, it enabled verification that performance remained saturated despite realistic sensor artefacts, thereby strengthening the claim that perception is not the mission bottleneck in this environment. Second, it facilitated targeted stress testing of the perception stack without requiring algorithmic modification.

Table 4. AprilTag detection probability, p_{detect}, under the baseline and three perception stress test regimes. LL/UL = lower/upper limit of the Jeffreys interval.

Scenario	n	Estimate	LL	UL
All lights	1639	1	0.998	1
Aisle lights off	163	0.687	0.613	0.755
No lights	96	0	0	0.026
Occluded tags	125	0.432	0.348	0.520

To begin quantifying how quickly performance degrades away from the "easy" *All lights* baseline regime, three perception stress-test scenarios were conducted for the warehouse inspection mission in Isaac Sim, and the AprilTag detection probability, p_{detect}, was reported for each scenario. The results are presented in Table 4. The first test, *aisle lights off*, disabled all the lights above the warehouse aisles, leaving only the southern warehouse lights on. This resulted in a reduction of detection probability from 1 in the baseline scenario, to $0.687\ [0.613,\ 0.755]$. Under the *no lights* regime, where all warehouse lights were turned off, performance collapsed to $0\ [0,\ 0.026]$. In

the *occluded tags* test, where tags were partially occluded by diagonal strips of hazard tape, p_{detect} degraded to 0.432 $[0.348, 0.520]$. Appendix B shows the *aisle lights off* and *occluded tags* test regimes. These demonstrations confirm that, under lower light levels and occlusion, the perception component becomes rate-limiting, and that Isaac Sim supports controlled, physics-based degradations that would be visible in the metrics. We note that the non-baseline sample sizes are modest, which explains the wider intervals and motivates larger sweeps.

From a dynamics perspective, Isaac Sim's contact and actuation fidelity plausibly contributed to the small but systematic differences observed between formal and simulated SoC costs (Table 3). The higher simulated `cost_06` ($+0.019$) is consistent with additional turning and settling overheads, short-horizon replanning, and path curvature that are captured by the k_v and k_ω terms in Equation (1), but would likely be under-represented in lower-fidelity simulators. Simulators with coarser contact and friction dynamics, alongside simplified local planning, can attenuate these overheads, resulting in optimistic energy budgets and consequently fewer early-dock events than would be expected on hardware. Therefore, the fact that the *docking low SoC* rate is higher in simulation than in the DTMC (by $+0.023$) is interpreted as a conservative result rather than a modelling artefact.

Integration and instrumentation were also differentiators. Isaac Sim's ROS 2 bridge and Isaac ROS perception nodes allowed running of the same Nav2 and AprilTag components as would be run on the physical robot, while logging precisely the events required for the PCTL-aligned metrics. The USD scene graph and programmatic control over materials, lights, and sensor rigs made it straightforward to hold nuisance variables fixed across 276 runs, while still enabling targeted perturbations. In contrast, reproducing camera-realistic lighting and sensor effects in Gazebo often requires custom plugins and hand-tuned noise models, which increase the risk that perception results reflect simulator-specific idiosyncrasies rather than physically grounded variation. This is visually apparent in the example Gazebo warehouse simulation assets shown in Appendix C.

Adopting Isaac Sim introduces trade-offs in computational demand, determinism, and workflow complexity. Achieving photorealistic sensing and high-fidelity dynamics required a highly capable GPU [19] and incurred higher start-up and iteration costs compared with a lightweight lower-fidelity workflow. This potentially restricts thorough exploration of the state space. Additionally, GPU-accelerated rendering can reduce strict determinism across runs unless seeds and scheduling are carefully controlled. The NVIDIA Omniverse and USD ecosystem also presents a steeper learning curve than legacy SDF-centric toolchains. Finally, while Isaac Sim helps reduce the perception domain gap, it does not eliminate it entirely. Factors such as camera response curves, sensor readout timing, and scene-specific material properties still require calibration if absolute accuracy is needed.

Overall, the findings suggest that the choice of simulator impacts the strength of evidence in CV&V for tasks that couple navigation with vision. For “easy” perception regimes, both Gazebo and Isaac Sim may yield similar top-line CV&V metrics. However, Isaac Sim should provide stronger assurance that the metrics will persist under realistic visual conditions, as well as more accurate estimates for energy- and recovery-related properties. For more challenging perception regimes, the gap is expected to widen in favour of Isaac Sim, as photorealistic rendering enables controlled, physics-based degradation studies that are not easily replicated with rasterised renderers.

5 Conclusions and Future Work

5.1 Conclusions

This work demonstrated a CV&V workflow that aligns a DTMC formal model with a high-fidelity Isaac Sim simulation for an autonomous warehouse inspection UGV. Across seven assurance metrics, the formal model’s reachabilities fall within the equal-tailed 95% Jeffreys intervals of the simulation estimates from 276 runs, providing formal-simulation CV&V for the mission and environment studied. The evidence shows that overall mission outcomes are dominated by navigation and resource management, while conditional perception performance remains saturated near unity under the evaluated, well-lit AprilTag conditions. Additional perception stress tests quantified the drop in p_{detect} under reduced illumination and occlusion, confirming that perception becomes rate-limiting under such degradations.

The analysis isolates the main sources of residual discrepancy. First, the simulation exhibits slightly higher rates of *docking low SoC* and marginally lower *all waypoints* success, consistent with additional turning and settling overheads as well as path curvature captured by the simulation’s energy model but under-represented in the DTMC’s per-segment cost abstraction. Second, a single *returned to dock* failure in simulation arose from concurrent docking triggers at the low-SoC threshold, motivating explicit arbitration of docking commands in both model and implementation. Third, fixed recovery costs in the DTMC reproduce top-level rates but cannot capture the observed variability across runs, suggesting a distributional treatment of recovery effort and state-dependent hazards.

A qualitative evaluation of Isaac Sim indicates that simulator fidelity materially affects the weight of evidence. Isaac Sim’s physically based rendering and contact dynamics support perception-in-the-loop testing with realistic sensor artefacts and more accurate energy and recovery estimates, thereby increasing confidence that the resulting assurance claims will generalise to hardware. These benefits come with trade-offs: higher computational requirements, stricter control of determinism, and a more complex workflow compared with lower-fidelity toolchains.

5.2 Future Work

Future work should strengthen the CV&V assurance case and broaden its applicability in four respects. First, incorporate a physical experimentation strand - for example, hardware-in-the-loop and on-robot trials - using the same metrics and logging schema so that experimental logs can serve as drop-in evidence. Such data should support calibration of scene material properties, frictional losses, SoC modelling, and camera response, thereby narrowing residual perception and energy gaps and enabling comprehensive, three-strand CV&V [3].

A second direction is to refine the formal abstraction. Recovery effort should be modelled as a distribution conditioned on local clutter or recovery duration; the model should include explicit arbitration between nominal and low-SoC docking triggers; per-leg costs should account for turning and approach overheads as well as curvature-dependent energy; and richer formulations, such as semi-Markov models [33] or reward-shaped variants [34], should be explored. In addition, the perception process should be represented explicitly via a scenario-conditioned detection parameter, with Bayesian calibration [35] and posterior predictive checks [36] against both baseline and stress-test runs.

A third avenue concerns expanded simulation scenarios. Building on the initial perception stress test demonstrations, the experimental programme should comprise systematic sweeps over lighting, occlusion, glare, motion blur, tag size, distance, and angle of incidence; increasing sample sizes to narrow Jeffreys intervals; and coupling perception outcomes to the defined CV&V metrics. Rare-event estimation via importance sampling [37] and other targeted rare-event methods [38] can be used to probe boundary cases. Cross-simulator ablations, including downshifted sensor models that mimic lower-fidelity pipelines, can quantify the sensitivity of the seven metrics to simulator choice, while counterexample-guided scenario generation from the formal model can further concentrate sampling on high-assurance regions.

Finally, process and reproducibility deserve attention. Extending the implemented goal structuring notation patterns to make energy- and perception-related claims explicit, automating metric extraction and traceability from runs to claims [39], and containerising seeds and configurations to control determinism will improve auditability [40], thereby raising the credibility and transferability of the overall CV&V approach.

References

[1] M. Fisher *et al.*, “An overview of verification and validation challenges for inspection robots,” *Robotics*, vol. 10, no. 2, p. 67, 2021 (cited on p. 8).

[2] D. B. Abeywickrama *et al.*, *Autonomy and safety assurance in the early development of robotics and autonomous systems*, 2025. arXiv: 2501.18448. [Online]. Available: https://arxiv.org/abs/2501.18448 (cited on p. 8).

[3] M. Webster *et al.*, “A corroborative approach to verification and validation of human–robot teams,” *The International Journal of Robotics Research*, vol. 39, no. 1, pp. 73–99, Nov. 2019, ISSN: 1741-3176. DOI: 10.1177/0278364919883338. [Online]. Available: http://dx.doi.org/10.1177/0278364919883338 (cited on pp. 8, 10, 11, 16, 22, 31).

[4] D. B. Abeywickrama *et al.*, *Autonomous robotic swarms: A corroborative approach for verification and validation*, 2025. arXiv: 2407.15475. [Online]. Available: https://arxiv.org/abs/2407.15475 (cited on pp. 8, 10).

[5] D. B. Abeywickrama *et al.*, “Corroborative V&V for Autonomous Systems: Integrating Evidence and Discrepancy Analysis for Safety Assurance,” *Agents and Robots for reliable Engineered Autonomy*, 2025, Unpublished manuscript (cited on pp. 8, 10, 12).

[6] M. Schwammberger *et al.*, *Integrating Formal Verification and Simulation-based Assertion Checking in a Corroborative V&V Process*, 2022. arXiv: 2208.05273. [Online]. Available: https://arxiv.org/abs/2208.05273 (cited on pp. 8, 10).

[7] R. C. Cardoso *et al.*, “Heterogeneous verification of an autonomous curiosity rover,” in *NASA Formal Methods*, R. Lee *et al.*, Eds., Cham: Springer International Publishing, 2020, pp. 353–360, ISBN: 978-3-030-55754-6 (cited on pp. 8, 10).

[8] NVIDIA, *Isaac Sim*, version 4.5.0. [Online]. Available: https://github.com/isaac-sim/IsaacSim (cited on pp. 8, 11–13, 27).

[9] M. Luckcuck *et al.*, “Formal specification and verification of autonomous robotic systems: A survey,” *ACM Computing Surveys*, vol. 52, no. 5, pp. 1–41, Sep. 2019, ISSN: 1557-7341. DOI: 10.1145/3342355. [Online]. Available: http://dx.doi.org/10.1145/3342355 (cited on p. 9).

[10] M. Webster *et al.*, “Generating certification evidence for autonomous unmanned aircraft using model checking and simulation,” *Journal of Aerospace Information Systems*, vol. 11, pp. 258–279, May 2014. DOI: 10.2514/1.I010096 (cited on p. 10).

[11] N. Koenig and A. Howard, “Design and use paradigms for gazebo, an open-source multi-robot simulator,” in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*, vol. 3, 2004, pp. 2149–2154. DOI: 10.1109/IROS.2004.1389727 (cited on p. 10).

[12] E. Berrocal, B. Sierra, and H. Herrero, “Evaluating pybullet and isaac sim in the scope of robotics and reinforcement learning,” in *2024 7th Iberian Robotics Conference (ROBOT)*, 2024, pp. 1–6. DOI: 10.1109/ROBOT61475.2024.10797383 (cited on p. 11).

[13] T. Erez, Y. Tassa, and E. Todorov, “Simulation tools for model-based robotics: Comparison of bullet, havok, mujoco, ode and physx,” in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 4397–4404. DOI: 10.1109/ICRA.2015.7139807 (cited on p. 11).

[14] Q. Le Lidec *et al.*, “Contact models in robotics: A comparative analysis,” *IEEE Transactions on Robotics*, vol. 40, pp. 3716–3733, 2024. DOI: 10.1109/TRO.2024.3434208 (cited on p. 11).

[15] C. Duarte *et al.*, “A quantitative comparison of vision performance for the hsr: Gazebo vs. isaac simulator,” in *Robot World Cup*, Springer, 2024, pp. 339–350 (cited on p. 11).

[16] A. Dosovitskiy *et al.*, “CARLA: An open urban driving simulator,” in *Proceedings of the 1st Annual Conference on Robot Learning*, 2017, pp. 1–16 (cited on p. 11).

[17] S. Shah *et al.*, *Airsim: High-fidelity visual and physical simulation for autonomous vehicles*, 2017. arXiv: 1705.05065. [Online]. Available: https://arxiv.org/abs/1705.05065 (cited on p. 11).

[18] NVIDIA, *NVIDIA PhysX*, version 5.6. [Online]. Available: https://github.com/NVIDIA-Omniverse/PhysX (cited on p. 11).

[19] Isaac Sim Documentation. “Isaac Sim Requirements,” NVIDIA. (Aug. 2025), [Online]. Available: https://docs.isaacsim.omniverse.nvidia.com/latest/installation/requirements.html (cited on pp. 11, 20).

[20] T. W. Kim *et al.*, “Surgical robotics environment in nvidia isaac sim for robot-assisted suturing,” in *2025 International Symposium on Medical Robotics (ISMR)*, 2025, pp. 192–198. DOI: 10.1109/ISMR67322.2025.11025977 (cited on p. 11).

[21] D. B. Abeywickrama *et al.*, “Towards patterns for a reference assurance case for autonomous inspection robots,” in *2025 IEEE/ACM 22nd International Conference on Software and Systems Reuse (ICSR)*, 2025, pp. 95–100. DOI: 10.1109/ICSR66718.2025.00016 (cited on p. 12).

[22] C. Menghi *et al.*, “Specification patterns for robotic missions,” *IEEE Transactions on Software Engineering*, vol. 47, no. 10, pp. 2208–2224, 2021. DOI: 10.1109/TSE.2019.2945329 (cited on p. 12).

[23] Segway Robotics, “Nova Carter Product Manual,” Segway Robotics and NVIDIA, Tech. Rep., 2023. [Online]. Available: https://robotics.segway.com/wp-content/uploads/2023/11/Nova-Carter-Product-Manual-v1.0.pdf (cited on pp. 12, 13).

[24] S. Macenski *et al.*, “Robot operating system 2: Design, architecture, and uses in the wild,” *Science Robotics*, vol. 7, no. 66, 2022. DOI: 10.1126/scirobotics.abm6074. [Online]. Available: https://www.science.org/doi/abs/10.1126/scirobotics.abm6074 (cited on p. 12).

[25] Open Navigation LLC, *Detailed behavior tree walkthrough: Navigate to pose with replanning and recovery*, https://docs.nav2.org/behavior_trees/overview/detailed_behavior_tree_walkthrough.html#navigate-to-pose-with-replanning-and-recovery, Nav2 documentation. Accessed 2025-08-15 (cited on pp. 12, 15).

[26] J. Wang and E. Olson, “Apriltag 2: Efficient and robust fiducial detection,” in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016, pp. 4193–4198. DOI: 10.1109/IROS.2016.7759617 (cited on pp. 12, 15, 19).

[27] NVIDIA, *NVIDIA Isaac ROS*, version 3.2. [Online]. Available: https://github.com/NVIDIA-ISAAC-ROS (cited on pp. 12, 18).

[28] M. F. Jaramillo-Morales *et al.*, “Energy estimation for differential drive mobile robots on straight and rotational trajectories,” *International Journal of Advanced Robotic Systems*, vol. 17, no. 2, 2020. DOI: 10.1177/1729881420909654 (cited on p. 13).

[29] M. Kwiatkowska, G. Norman, and D. Parker, “PRISM 4.0: Verification of probabilistic real-time systems,” in *Proc. 23rd International Conference on Computer Aided Verification (CAV’11)*, G. Gopalakrishnan and S. Qadeer, Eds., ser. LNCS, vol. 6806, Springer, 2011, pp. 585–591 (cited on pp. 14, 31).

[30] E. Innocenti, G. Agostini, and R. Giuliano, “Uavs for medicine delivery in a smart city using fiducial markers,” *Information*, vol. 13, no. 10, 2022, ISSN: 2078-2489. DOI: 10.3390/info13100501. [Online]. Available: https://www.mdpi.com/2078-2489/13/10/501 (cited on p. 15).

[31] S. Macenski *et al.*, “The marathon 2: A navigation system,” in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 2718–2725. DOI: 10.1109/IROS45743.2020.9341207 (cited on pp. 15, 17).

[32] L. D. Brown, T. T. Cai, and A. DasGupta, “Interval estimation for a binomial proportion,” *Statistical Science*, vol. 16, no. 2, pp. 101–117, 2001, ISSN: 08834237, 21688745. DOI: 10.1214/ss/1009213286. [Online]. Available: http://www.jstor.org/stable/2676784 (visited on 08/24/2025) (cited on pp. 15, 16).

[33] R. S. Sutton, D. Precup, and S. Singh, “Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning,” *Artificial Intelligence*, vol. 112, no. 1, pp. 181–211, 1999, ISSN: 0004-3702. DOI: https://doi.org/10.1016/S0004-3702(99)00052-1. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0004370299000521 (cited on p. 22).

[34] A. Y. Ng, D. Harada, and S. J. Russell, “Policy invariance under reward transformations: Theory and application to reward shaping,” in *Proceedings of the Sixteenth International Conference on Machine Learning*, ser. ICML ’99, San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1999, pp. 278–287, ISBN: 1558606122 (cited on p. 22).

[35] M. C. Kennedy and A. O’Hagan, “Bayesian calibration of computer models,” *Journal of the Royal Statistical Society Series B: Statistical Methodology*, vol. 63, no. 3, pp. 425–464, Jan. 2002, ISSN: 1369-7412. DOI: 10.1111/1467-9868.00294. [Online]. Available: https://doi.org/10.1111/1467-9868.00294 (cited on p. 22).

[36] A. Gelman, X.-L. Meng, and H. Stern, “Posterior predictive assessment of model fitness via realized discrepancies,” *Statistica Sinica*, vol. 6, no. 4, pp. 733–760, 1996, ISSN: 10170405, 19968507. [Online]. Available: http://www.jstor.org/stable/24306036 (visited on 08/31/2025) (cited on p. 22).

[37] S. Juneja and P. Shahabuddin, “Rare-event simulation techniques: An introduction and recent advances,” *Handbooks in operations research and management science*, vol. 13, pp. 291–350, 2006. DOI: https://doi.org/10.1016/S0927-0507(06)13011-X (cited on p. 22).

[38] X. Chen *et al.*, *Understanding domain randomization for sim-to-real transfer*, 2022. arXiv: 2110.03239. [Online]. Available: https://arxiv.org/abs/2110.03239 (cited on p. 22).

[39] D. B. Abeywickrama, *Ecv2: Eclipse plugin for corroborative verification and validation for autonomous systems*. [Online]. Available: https://github.com/DhamindaA/ecv2-eclipse-plugin (cited on p. 22).

[40] C. Boettiger, “An introduction to docker for reproducible research,” *SIGOPS Oper. Syst. Rev.*, vol. 49, no. 1, pp. 71–79, Jan. 2015, ISSN: 0163-5980. DOI: 10.1145/2723872.2723882. [Online]. Available: https://doi.org/10.1145/2723872.2723882 (cited on p. 22).

[41] AWS Robotics, *AWS Robomaker small warehouse world*, Dec. 2021. [Online]. Available: https://github.com/aws-robotics/aws-robomaker-small-warehouse-world (cited on p. 30).

[42] Open Robotics, *Warehouse*, Sep. 2023. [Online]. Available: https://fuel.gazebosim.org/1.0/OpenRobotics/models/Warehouse (cited on p. 30).

Appendices

A Warehouse Environment

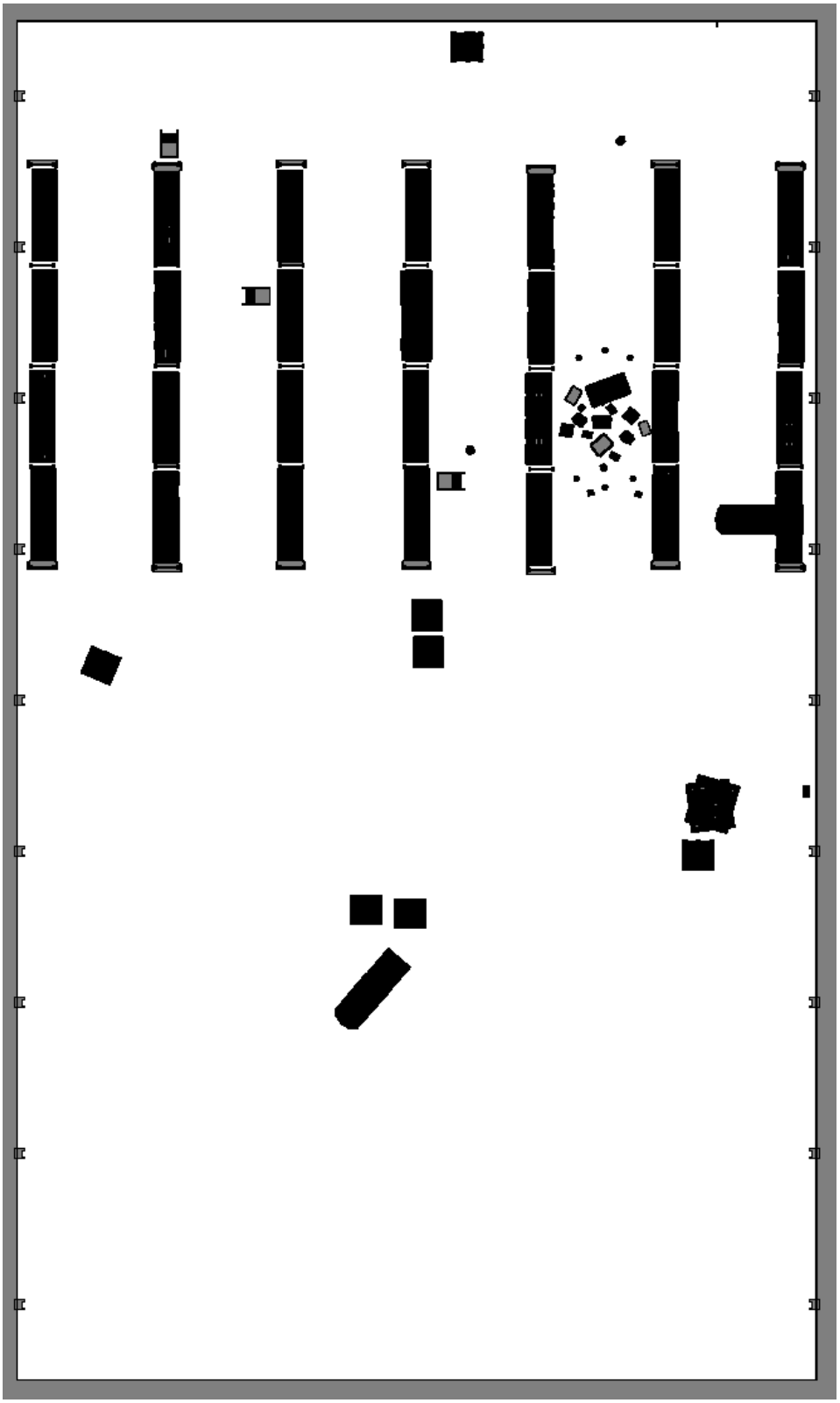


Fig. 1. Occupancy map at the height of the Carter's front lidar ($z = 0.62$ m) of the adapted 54×32 m `full_warehouse.usd` environment asset [8]. Occupied areas are shown in black, unknown areas in grey, and free areas in white.

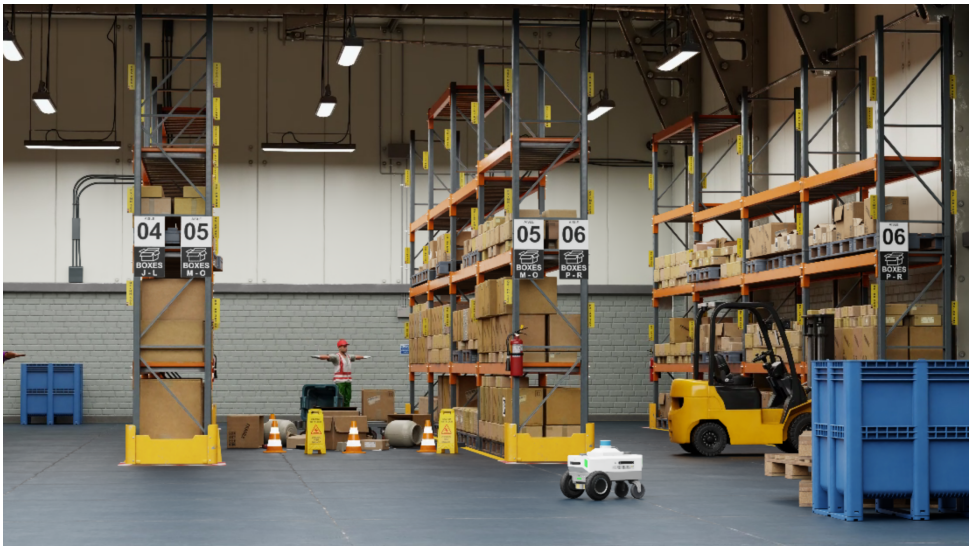
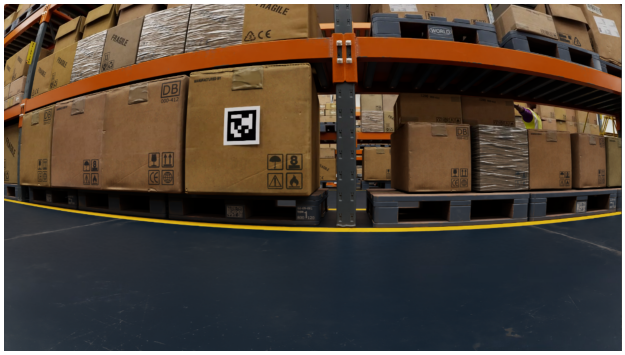


Fig. 2. View of the Nova Carter P1 UGV moving in the photorealistic warehouse environment.

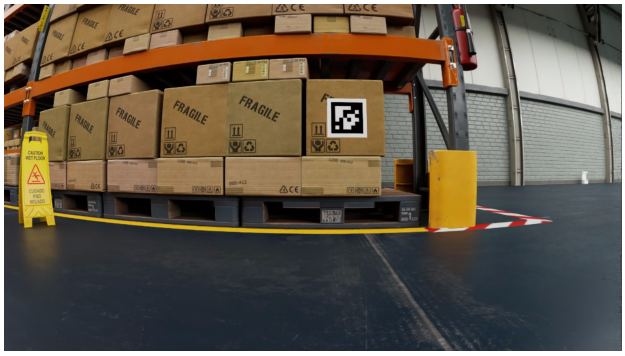
(a) Waypoint 1

(b) Waypoint 2

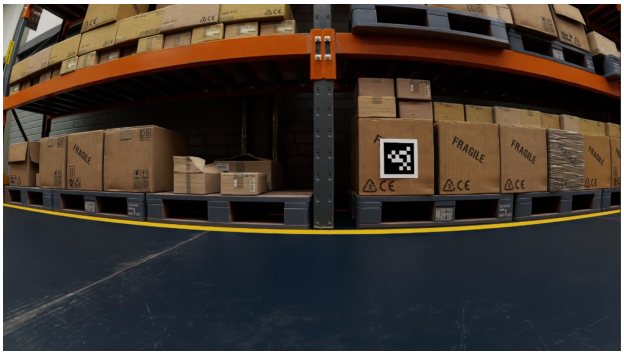


(c) Waypoint 3

(d) Waypoint 4



(e) Waypoint 5



(f) Waypoint 6

Fig. 3. Per-waypoint views out of Nova Carter's front Hawk stereo camera showing tag36h11 AprilTag markers placed on crates or boxes, with unique IDs corresponding to the waypoint number.

B Perception Stress Tests

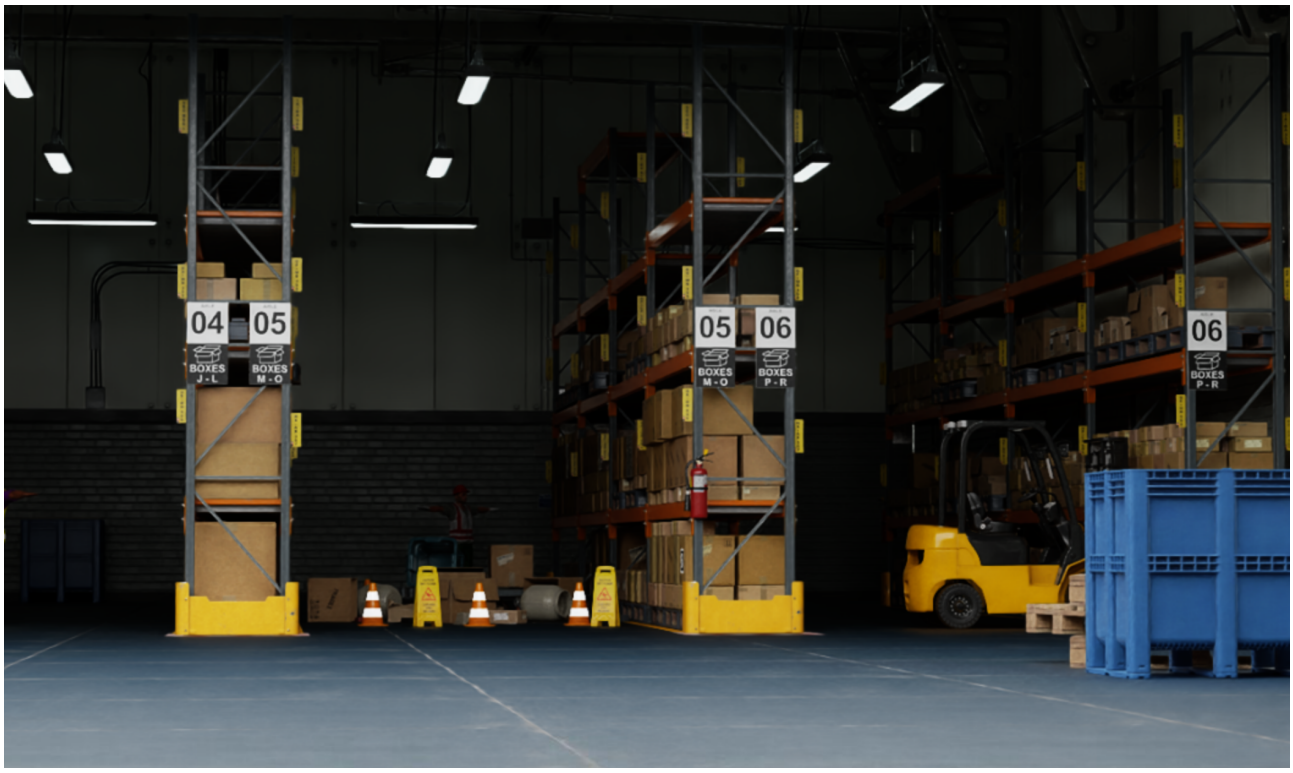


Fig. 4. View of the warehouse in the *Aisle lights off* perception stress test.

Fig. 5. View of waypoint 1 in the *occluded tags* perception stress test. The AprilTag has been partially occluded by diagonal strips of hazard tape.

C Example Gazebo Warehouse Environments

Fig. 6. Example warehouse environment in Gazebo created by AWS Robotics [41].

Fig. 7. Example warehouse environment in Gazebo created by Open Robotics [42].

D Project Outline

D.1 Scope

This dissertation investigates how NVIDIA Omniverse - specifically Isaac Sim - can be integrated into a Corroborative Verification and Validation (CV&V) [3] workflow for an autonomous inspection robot operating in a representative industrial facility. The study will:

1. Design and simulate an individual ground-based inspection robot in Isaac Sim. The robot shall be capable of patrolling the area and taking measurements at prescribed waypoints while remaining inside a predefined safe zone and avoiding collisions with static obstacles.
2. Construct a formal probabilistic model of the mission using a model checker such as PRISM [29]. The model shall ideally incorporate motion uncertainty, sensor errors, and single-fault assumptions.
3. Corroborate the results from Isaac Sim and the PRISM model to build assurance arguments for (a) safety, shown by the robot staying within the geofenced safe zone and avoiding collisions, and (b) mission success, shown by the completion of every required measurement.

No physical robots will be built or deployed. Compute will be limited to a single remote NVIDIA L40S GPU, available weekdays 08:00 - 18:00. The outcome will include reproducible Universal Scene Descriptor (USD) environments, a formal probabilistic model, and evaluation metrics demonstrating strengths, limitations, and recommended practices for Omniverse-centric CV&V of autonomous robotic inspection.

D.2 Motivation

Digital twin fidelity now enables simulated robots to meaningfully substitute for hardware during early-stage CV&V, reducing both cost and risk. Yet most CV&V pipelines still pivot around legacy simulators that compromise visual realism, sensor noise modelling, and reproducibility. NVIDIA Omniverse offers a scalable, photorealistic alternative that promises to close these gaps via GPU-accelerated simulation and USD-centric asset management. Demonstrating Omniverse's suitability for a probabilistic assurance case of an autonomous inspection robot will provide concrete evidence and actionable workflows for adopting a digital-world backbone through the CV&V lifecycle. This addresses an urgent industrial and academic need for more rigorous, transparent, and tool-agnostic CV&V of increasingly complex autonomous systems.

D.3 Aims and Objectives

#	Aim/Objective	Success Criteria
O1	Complete Isaac Sim, USD, and PRISM upskilling.	Completion of official tutorials; working demo robot in Isaac Sim; verified PRISM toy model.
O2	Implement inspection mission in Omniverse.	USD simulation of a robot taking measurements at waypoints in a warehouse within a defined safe area while avoiding collisions with static obstacles.
O3	Develop formal probabilistic model of the inspection task in PRISM.	Model compiles; probabilistic reachability and safety queries return within accepted bounds.
O4	Integrate results in a corroborative pipeline.	Evidence showing convergence/divergence of results across methods.
O5	Quantitatively assess Omniverse's impact on CV&V quality and effort.	Comparative statistics vs. legacy Gazebo implementation as described in literature.
O6	Publish open-sourced, reproducible assets.	Public Git repository with automated build/test and reproduction manual.

D.4 Project Plan

Week	Planned Activities
1	Set up environment; Familiarisation with Omniverse; Project plan
2	Literature review; Scoping of title
3	Isaac Sim upskilling; Project outline; Risk Assessment
4	Design inspection mission spec; Prototype single-robot navigation in a simple arena
5	Develop PRISM abstraction; Validate against simplified model
6	Build data collection and logging pipeline
7	Execute parameter sweeps in Isaac Sim
8	Run PRISM experiments; Perform sensitivity analysis; Begin convergence study
9	Draft results section; Generate visualisations
10	Draft discussion section; Create public Git repository
11	Implement supervisor feedback; Full dissertation draft
12	Final proofreading and revisions; Contingency buffer

E Risk Assessment

Risk Assessment

Date:	Assessed by:	Approved by:	Building / Location:	Assessment ref no:	Review date:
26/06/25	Euan Baldwin		Nancy Rothwell		01/09/25

Task/Premises:
MSc Robotics Dissertation

Activity	Hazard	Who might be harmed and how	Existing measures to control risk	Risk rating	Result
Working on campus	COVID infection through close contact or the contact with surfaces which may have been contaminated by previous users	Staff, students, visitors Infection of respiratory illness	• COVID restrictions have ceased in the UK. • Face coverings and hand sanitisers remain available at main entrances of University buildings. • For the latest University's COVID guidance, please see StaffNet https://www.staffnet.manchester.ac.uk/campus-management	Med	A

Activity	Hazard	Who might be harmed and how	Existing measures to control risk	Risk rating	Result
Working on campus	Building fire	Staff, students, visitors If present within the building during a fire Burns, Smoke inhalation	• Induction arrangements cover security and fire awareness and include how to locate and use a fire door to exit the building and the location of the fire assembly point(s). • All new staff should complete fire awareness e-training TLCF100. • Fire Action notices are displayed around the building • Fire alarm system are in place and tested weekly on day at time to enable users to identify the sound of the alarm, see fire action notice at entrance to buildings. • Fire evacuation practices are carried out annually as a minimum • Building users are empowered to activate the fire alarm if a building evacuation is necessary during an emergency • Induction covers the importance of maintaining clear fire exit routes and keeping the doors closed unless essential. Induction also covers the need for high general housekeeping standards. • Ready access to fire extinguishers is available for use by trained users. • Staff 'hosts' are responsible for the safety and evacuation of visitors. • Evacuation marshals attend suitable training and assist where possible during evacuations during normal working hours. • Requests to work out of hours include emergency action in case of fire and use of fire routes and doors.	Med	A
Working on campus	Injuries or ill health	Staff, students, visitors	• First aiders are available and First Aid Notices are displayed around the building • All Campus Security staff are first aid trained. Security contact details are 0161-306-9966. This telephone number can be found on the back of staff/student ID cards. • AEDs/ Defibrillators are located throughout campus, please see map for nearest location	Med	A
Maintaining building security	Suspicious people/ activities in and around campus	Staff, students, visitors Difficulty in contacting help/assistance	• If using a swipe card to access a building, do not allow anyone to tailgate • If you see any suspicious activities in and around the premises, get yourself to a safe place and call Campus Security immediately on 0161 306 9966	Med	A

Activity	Hazard	Who might be harmed and how	Existing measures to control risk	Risk rating	Result
			• Do not enter into any area unauthorised for your use, lone working or out-of-hours • Do not prop doors open • When entering and exiting the building, keep to well-lit area and be extra vigilant of surroundings		
Working on campus and traversing around the buildings	Building defects and poor housekeeping	Staff, students and visitors. Discomfort while working and physical injuries if building defects cause an accident Slips, trips and falls	• Defects or concerns can be reported to Estates Helpdesk by calling 0161 275 2424 or using the on-line reporting form Estates Helpdesk • Reasonable standards of housekeeping should be maintained and checked on regularly. • Floors kept clean, dry and clear of obstructions particularly exit routes. Spillages to be cleared immediately • Cabinet drawers and doors are kept closed when not in use. Items should be stored securely to avoid items falling or people colliding with protruding items. • Trailing cables must be positioned neatly away from walkways or secured and highlighted with hazard tape. • Fan heaters or air conditioning units should not be brought into the space unless facilitated by Estates. • Waste bins are supplied for general and recyclable waste reducing the build-up of rubbish in corridors and spaces. • All communal spaces should be treated with respect and House services will be conducting regular cleaning of these spaces. • Adequate lighting is based on identified activities/tasks in the areas as deemed sufficient during building design specification. • Emergency lighting will turn on if standard lighting system is faulty to ensure there will always be light in the areas.		
Allocation of workspace	Lack of space impacting safe access and egress	Staff, students, visitors Physical injuries and obstruction of access and egress	• Use appropriately sized furniture. • Reasonable standards of housekeeping are maintained and checked on regularly by users. • Adequate space should be maintained between furniture and/or items to permit easy access and egress. • Stairwells, corridors, fire escapes and circulation routes are not to be used as storage space and must be kept clear at all times.	Low	A

Activity	Hazard	Who might be harmed and how	Existing measures to control risk	Risk rating	Result
Using kitchen facilities	Inadequate maintenance of water cooler or water dispenser giving rise to Legionella	Staff, students, visitors Legionnaires disease could be contracted from inhalation of water aerosols	• Estates maintain the mains fed water coolers and local areas oversee the stand-alone water coolers. • In both cases above regular maintenance should be carried out and records kept. This is usually achieved via service contracts.	Low	A
Using kitchen facilities	Poor waste management/ disposal	Staff and students, visitors. Discomfort from poor housekeeping and odours	• A variety of waste bins/streams are supplied for recycling and disposal needs. • House services staff dispose of general waste regularly. • Food waste should be disposed of immediately to maintain hygiene and avoid vermin. • Food items should be stored correctly to avoid vermin. • Clean up spillages.	Low	A
Use of kitchen appliances	Kitchen appliances such as kettles, fridges, microwaves and dishwasher Misuse of electrical equipment, hot food and liquids, microwave radiation leakage, poor hygiene	Staff, students, visitors Electric shocks, fire, burns, scalding, treatment for microwave radiation leakage	• Appliances are PAT tested regularly. If faulty, stop use immediately and report it. • All kitchen users are responsible for keeping these appliances and kitchen areas clean and free from spillages. • Cleaning materials should be stocked in the kitchens. Kettle • When using a kettle, check the water level is correct. Do not overfill. Do not let it boil dry. • Position the cups so they are near the kettle to reduce the distance required to pour the hot water. • Pay attention when pouring the hot water to avoid spillage. Microwave • Only use microwave-safe containers in a microwave • When using a microwave, ensure the correct temperature and duration are used. Do not leave food heating unattended in microwave. • Be mindful of steam when opening the microwave door. Keep a safe distance. • Protect your hands when taking hot food out of the microwave, e.g. use a tea towel Fridges • All kitchen users are responsible for disposing of their own unwanted food items. Do not leave food and drinks to go out of date or become rotten in the fridge • If possible, place loose food items in secure containers	Low	A

Activity	Hazard	Who might be harmed and how	Existing measures to control risk	Risk rating	Result
			Dishwasher • Follow the user manual • Only put dishwasher-safe items into the dishwasher • Only use dishwasher tablets and wash hands immediately after touching the tablets. • Do not overfill a dishwasher • Place items in a secure position in the dishwasher. Do not allow fragile items to fall over • Only open the dishwasher once the washing cycle has finished and it has cooled down • Be mindful of steam when opening the dishwasher door. Keep a safe distance		
Use of office electrical equipment, both Personal and University Owned	Electric shocks Fire Damage to other electrical equipment Misuse of electrical appliance, faulted electrical appliance.	Staff, students, visitors Burns, Smoke inhalation,	• All University electrical equipment will undergo Portable Appliance Testing. • Staff are discouraged from bringing in own electrical equipment as maintenance cannot be assured. • Personal Equipment will also need to undergo portable appliance testing before use within UoM buildings. • Any damaged equipment should be taken out of service and either replace or repaired. • All equipment whether personal or UoM owned must comply with relevant standards such as the British Standard or EU standards. • All equipment should be used in accordance with the manufacturer's instructions. • Liquid spills near electrical equipment should be cleaned up immediately. • Extension cables should be avoided as much as possible. Daisy-chaining is not permitted. • Visual checks before use to make sure equipment, cables and free from defects • Defective plugs, cables equipment etc. should be taken out of use and be reported for repair/replacement.	Low	A
Use of display screen equipment Repetitive/prolonged use of equipment or tasks	Incorrect posture whilst using DSE Incorrect workstation set up	Staff, students, visitors Musculoskeletal injuries/disabilities Limb disorders	• Please refer to the DSE policy, guidance and poster for more information on how to set up your workstation properly	Low	A

Activity	Hazard	Who might be harmed and how	Existing measures to control risk	Risk rating	Result
	Prolonged use without breaks Electrical hazards	Eye strain Headaches Back pain Repetitive strain Fatigue Electric shock	• Complete DSE Self-Assessment for a Safety Advisor to review and report back with any recommendations or actions. • Seats should be stable and adjustable to provide comfort • Set up workstation to a comfortable position with good lighting and natural light where possible • Take regular breaks away from the screen. • Regularly stretch your arms, back, neck, wrists and hands to avoid repetitive strain injuries. Refer to workstation exercises here • Provision of adjustable equipment and furniture available following DSE assessment • Refer to use of electrical equipment. • Any work of a repetitive nature must be subject to a separate risk assessment in consultation with a Safety Advisor		
Activity on your own	Lone working	Staff, students, visitors Isolated, unable to summon assistance	• Please refer to the University Lone Working policy and guidance for more information • Please refer to the new University Working at Home guidance • Please refer to the new University Wellbeing Support website • Please refer to the FSE Personal Safety Guidance • Staff to remain in regular contact with line manager or colleagues via Teams, Zoom, phone or email etc.	Low	A

Activity	Hazard	Who might be harmed and how	Existing measures to control risk	Risk rating	Result
Working out of hours	Potential for lone-working Changes to the environment during evenings and weekends	Staff, students, visitors More vulnerable. Difficulty in contacting help/assistance	• Out of hours working to be approved by line manager/ Academic Supervisor beforehand. • Minimise the duration and frequency of working out of hours. • Carry a charged up mobile phone on person at all times. • Be aware of out of hours safety protocols, including security contact telephone numbers, evacuation and first aid information. • General building and campus support will be reduced out of hours. • Inform someone beforehand of the planned lone working (time, location and duration). Set up a buddy system so you contact someone at regular intervals (within the building if possible or by telephone /emails/ Teams etc.) • Accompanied buddy is for high-risk activities = Work with another person in the same area in close proximity • Remote buddy is for low-risk activities = Regular contact with another person via visits, phone, texts or emails • SafeZone app can be set with a check-in timer during out of hours use. Should the timer not be switched off, security and/ or remote buddy will be alerted to call occupant. • In an emergency or if in need of first aid call Campus Security on 0161 3069966	Med	A
Work pressures	Stress	Staff, students, visitors Stress related illness (causes may include: pressure of work, insufficient support from colleagues/line management)	• Please refer to Stress Prevention and Management toolkit for policies and guidance • Please refer to Guidance for Managers and Guidance for Staff • Complete training Work Related Stress: Identification, Prevention & Management (Online) • The University Stress Assessment tool can be used to highlight the main factors for an individual that are recognised as having the potential to lead to work-related stress • Projects, work plans and objectives to be discussed and agreed at annual PDR or more frequently if required. • Refer to full FSE Stress Risk Assessment • Regular contact meetings with manager and peers, Skype, Zoom, Phone	Low	A

Activity	Hazard	Who might be harmed and how	Existing measures to control risk	Risk rating	Result
			• Define working hours, set a start & close daily routine, and prioritise your tasks. • Individual may self-refer to Occupational Health Service or to the Counselling and Mental Health Service		

I confirm that I have read this Risk Assessment and that I understand the hazards and risks involved and will follow all of the safety procedures stated.

Name (please print)	Signed	Line manager /PI countersignature	Date
Euan Baldwin	Euan Baldwin		26/06/25