Bridging Research and Practice in Simulation-based Testing of Industrial Robot Navigation Systems

Sajad Khatiri*, Francisco Eli Viña Barrientos†, Maximilian Wulf‡, Paolo Tonella§, Sebastiano Panichella¶
*§Università della Svizzera italiana, Lugano, Switzerland
*¶University of Bern, Bern, Switzerland
†‡ANYbotics AG, Zurich, Switzerland
sajad.mazraehkhatiri@unibe.ch, {fvina, mwulf}@anybotics.com, paolo.tonella@usi.ch, sebastiano.panichella@unibe.ch

***Abstract*—Ensuring robust robotic navigation in dynamic environments is a key challenge, as traditional testing methods often struggle to cover the full spectrum of operational requirements. This paper presents the industrial adoption of Surrealist, a simulation-based test generation framework originally for UAVs, now applied to the ANYmal quadrupedal robot for industrial inspection. Our method uses a search-based algorithm to automatically generate challenging obstacle avoidance scenarios, uncovering failures often missed by manual testing. In a pilot phase, generated test suites revealed critical weaknesses in one experimental algorithm (40.3% success rate) and served as an effective benchmark to prove the superior robustness of another (71.2% success rate). The framework was then integrated into the ANYbotics workflow for a six-month industrial evaluation, where it was used to test five proprietary algorithms. A formal survey confirmed its value, showing it enhances the development process, uncovers critical failures, provides objective benchmarks, and strengthens the overall verification pipeline.**

***Index Terms*—Robotic Navigation, Local Planning, Obstacle Avoidance, Simulation Environments, Search-Based Testing, Environment Generation, Quadrupedal Robots**

I. Introduction

Robots are revolutionizing industrial workflows in manufacturing, logistics, inspection, and maintenance by improving efficiency, productivity, and safety [1], [2]. Mobile autonomous robots are key for operating in complex, dynamic environments and handling hazardous or impractical tasks [3]–[7]. This trend demands robust, reliable platforms for safe real-world deployment [8], [9]. Autonomous navigation [10]—the ability to plan and follow paths—is vital for industrial robots. Obstacle avoidance enables safe interaction with static and dynamic hazards [11], including equipments, vehicles, and humans [12]–[14]. Failures can lead to collisions, damage, downtime, and safety risks [15], [16], underscoring the need for rigorous testing [9], [17]–[19].

Robotic navigation testing traditionally relies on manual design and physical trials [20]–[22], which are costly, time-consuming, and fail to cover diverse real-world scenarios [9], [23]–[25], especially unpredictable obstacle interactions [26]. Physical testing is essential but risky for edge cases, making comprehensive scenario coverage and robust validation difficult. Simulation-based testing offers a flexible, effective alternative to traditional methods [18], [27], [28]. High-fidelity simulators accurately model robots, sensors, and environments, enabling safe, repeatable, and cost-effective testing of navigation and control. They also facilitate systematic exploration of the operational design domain, including rare or hazardous edge cases [9], [29], [30].

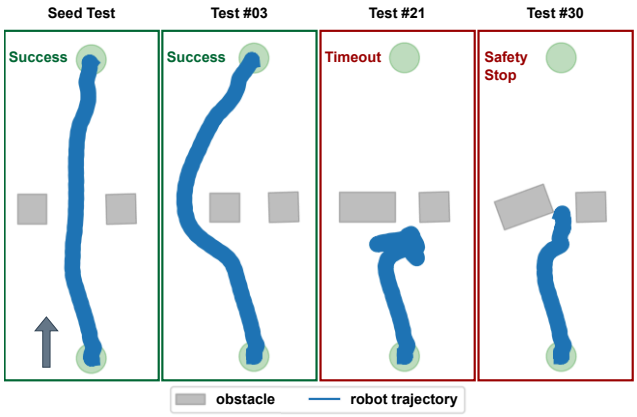

Fig. 1: Test generation process using Surrealist for a 2D robot navigation task, where the robot navigates to a target pose (x, y position and yaw angle). Starting with a manually defined seed test containing two obstacles (left), the system iteratively modifies the scenario by moving, resizing, and rotating the left obstacle to actively induce failures of the navigation algorithm. This uncovers failure cases that are often missed by manual or purely randomized testing.

In prior work on aerial robotics [9], [17], [31]–[34], we presented two frameworks: *Surrealist* [9], a search-based tool that uses evolutionary algorithms to generate realistic UAV simulation tests, and *Aerialist* [17], a PX4-based UAV test bench. *Surrealist* uncovered critical PX4 [35], [36] failure cases involving obstacle avoidance. Here, we extend them to a state-of-the-art quadrupedal robot to evaluate the benefits of automated test generation in an industrial workflow.

This study was conducted in close collaboration with *ANYbotics*[1], a world leader in developing quadrupedal robots for autonomous inspection and monitoring in industrial environments. The evaluation used their flagship platform *ANYmal D* [37], shown in Figure 2. The integration of our tools consisted of a pilot and deployment phase. During the pilot, we adapted the *Surrealist* [9] and *Aerialist* [17] frameworks to test quadrupedal navigation algorithms (see Figure 3). The adapted approach was initially evaluated using two exper-

[1]https://www.anybotics.com/

2643-1572/25/$31.00 ©2025 IEEE
DOI 10.1109/ASE63991.2025.00277

imental navigation algorithms (*Exp-Nav-A* and *Exp-Nav-B*) as test subjects and proved highly effective; for instance, test suites automatically generated in five scenarios revealed critical weaknesses in *Exp-Nav-A*, leading to a mission success rate of just 40.3%. These challenging test suites also provided an effective benchmark, demonstrating the superior robustness of *Exp-Nav-B*, which achieved a 71.2% success rate under the same conditions. Figure 1 illustrates examples from the test generation process for *Exp-Nav-B*.

After a successful pilot, the *ANYbotics* team integrated the framework into their development workflow. Over a period of six months, they first generated three test suites for their current released version (*ANY-NAV-A*), assessed its performance, and identified its core deficiencies to improve in the next version. The same test suites were used to benchmark four internal candidates for the next release (*ANY-NAV-*B_{1-4}), allowing them to effectively compare algorithms and iterate faster. While performance details remain confidential, feedback from a formal survey completed by eight engineers confirmed the framework's effectiveness. Specifically, engineers reported that "the framework significantly streamlined their workflow by automating challenging test generation, proved effective at uncovering algorithm deficiencies missed by manual methods, and enhanced their ability to objectively benchmark algorithm improvements." While the evaluation centered on the ANYmal quadruped, the methodology is generalizable to other mobile robots.

The main contributions of this paper are:

- Extension of the UAV-focused test generation framework (*Surrealist* [9], *Aerialist* [17]) to quadruped navigation—introducing challenges from high-dimensional locomotion, terrain contacts, and tightly coupled perception and planning in cluttered environments;
- Industrial evaluation at *ANYbotics AG* using the *ANYmal* quadruped over six months enabled engineers to efficiently: A) identify algorithm failures, B) iterate faster, and C) benchmark consistently;
- Empirical insights on challenges and trade-offs in simulation-based validation of industrial robotics, from a survey involving eight ANYbotics engineers.

The paper is organized as follows: Section II covers the ANYmal robot and testing frameworks. Sections III and IV detail our integration and methodology. Section V presents evaluation results, followed by discussion, threats, and related work in Sections VI and VII. We conclude and outline directions for future work in Section VIII.

II. Background

A. Robotic Locomotion and Navigation

Robotic mobility—the ability to move purposefully through the environment—is vital in applications such as industrial inspection, logistics, and search and rescue [1], [2]. It is typically tackled via a hierarchy: *locomotion* controls low-level actuators for stable movement, and *navigation* plans high-level paths to goals while avoiding obstacles [39]. Locomotion strategies vary with robot design and use [40]. Wheeled robots excel on smooth terrain, while tracked ones provide better mobility on uneven surfaces [41]. Aerial robots maneuver in 3D [41] and legged robots, especially quadrupeds, navigate stairs and rough terrain with diverse gaits like walking and trotting [39], [42], [43]. Stable, efficient movement relies on advanced control methods such as Model Predictive Control [44] and reinforcement learning [39], [43], [45].

Complimentary, navigation addresses the intelligent decision-making required for a mobile robot to reach a desired destination safely and efficiently [40]. Common navigation scenarios range from simple point-to-point movement to complex tasks like traversing a cluttered warehouse for a wheeled robot, inspecting a multi-story building with a drone, or navigating rough terrain with a quadrupedal robot. Robust navigation relies on Perception, integrating sensor data—cameras, LiDAR, and IMUs—to build an environmental model [46]. Navigation systems must plan optimal or near-optimal paths, avoid static and dynamic obstacles, and adapt to unforeseen environmental changes [46]. Planning typically involves *global planning*, which maps a high-level route to the goal, and *local planning*, which manages real-time obstacle avoidance [46]. For quadrupedal robots, navigation is closely tied to locomotion control, as gait and dynamic capabilities influence terrain traversal and obstacle avoidance [47].

With the rise of commercially available quadrupedal robots like ANYmal [37], Spot [48], Unitree Go [49], and Mini-Cheetah [50], research on navigation using these platforms has surged. Recent work spans reinforcement learning for terrain adaptation [45], dynamic maneuvers like obstacle jumping [51], and robust navigation via hierarchical state estimation [47]. Efforts also address safe operation in dynamic settings using control barrier functions [52], anisotropy-aware planning [53], and combining locomotion with high-level perception for complex tasks like parkour [42]. Human-robot safety is also being enhanced through visual tracking and predictive control [54]. These advances would have been impossible without systematic simulation-based testing, which remains essential for ensuring robustness and reliability.

B. The ANYmal Quadrupedal Robot

The *ANYmal* robot, developed by *ANYbotics*, is a quadrupedal platform built for autonomous inspection and monitoring in complex industrial environments [37]. Tasks typically include navigating to designated locations, collecting sensor data (e.g., visual, thermal, gas), and detecting anomalies such as leaks, corrosion, or equipment faults [55]. Its legged design enables mobility over human-centric terrains like stairs.

Figure 2 shows *ANYmal D*, the latest commercial version, with key components and sensors. It includes a 360° LiDAR for mapping and obstacle detection, wide-angle front and rear cameras for navigation, and six depth cameras for all-around terrain perception [38]. These sensors provide rich data for robust obstacle avoidance. In this study, ANYmal represents advanced quadrupedal robots with dynamic locomotion, multi-modal perception, and autonomous navigation [37].

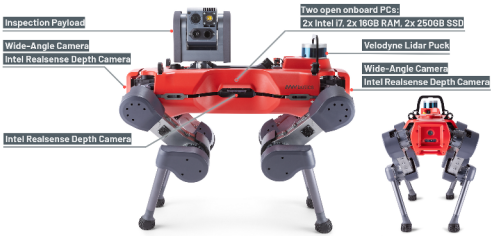

Fig. 2: ANYmal D architecture and sensors [38]

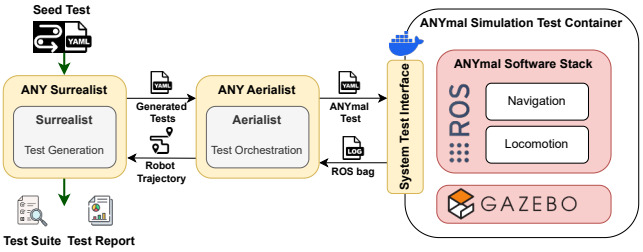

Fig. 3: Architecture of our simulation-based test generation framework

C. Surrealist and Aerialist Frameworks

Surrealist[2] [9] is a simulation-based test generation approach for UAVs that uses real flight logs to enhance the realism and effectiveness of testing. It operates in two phases: *replication*, which reconstructs real flights by optimizing obstacle configurations in simulation, and *generation*, which explores variations of the replicated environment to generate more challenging scenarios. These variations, primarily obstacle modifications, help reveal weaknesses or unsafe behaviors by pushing the system close to real-world limits. Surrealist leverages an evolutionary search-based algorithm to iteratively find obstacle properties with the best 'fitness' (detailed in [9]).

Built on the Aerialist[3] [17] test bench, Surrealist inherits capabilities for defining test cases, generating simulation configs, parallel test execution (via Kubernetes), and automated result analysis for PX4 firmware [35]. The versatility of Aerialist has established it as a foundational component in UAV test generation research [32]–[34]. Building on the success in the UAV domain, this paper adapts the combined search-based test generation framework to quadruped navigation, showing cross-domain applicability.

III. Integration Approach

We propose an end-to-end simulation-based test generation framework for quadruped navigation and obstacle avoidance. Our methodology integrates three key components (Figure 3): the *ANYmal* simulation test infrastructure, the *ANY Aerialist* test bench, and the *ANY Surrealist* test generator. The aim is to automatically generate challenging simulation-based test scenarios (including tricky obstacle configurations) to test ANYmal's autonomous navigation requirements, e.g., safety and reliability. *ANY Surrealist* generates test scenarios, *ANY Aerialist* executes them via a new ANYmal interface, and the results are analyzed for potential weaknesses. Yellow highlights in Figure 3 indicate integration extensions, detailed below.

A. System Under Test (ANYmal) Adaptations

ANYmal, like other complex robots, has a large codebase, advanced software architecture, and a sophisticated multi-computer deployment setup. To simplify integrating open-source test generation tools (Aerialist and Surrealist) with ANYmal and similar robots, we created the *System Test Interface*—a facade that abstracts software setup, simulation configuration, and test execution using our previously defined *test definition* [9], [17], which specifies the system, simulation, mission, and runtime commands.

To apply this to ANYmal, we built on its existing testing setup. ANYbotics provides a robust simulation-based infrastructure using the Gazebo simulator and a detailed ANYmal D model with full kinematics, dynamics, and sensors. We added a command line interface (CLI) to support external, automated scenario configuration and control. The core is an Aerialist test definition file (Figure 4, left), enabling precise specification of:

- *Obstacle Parameters:* number, size, shape, and position of static obstacles within the simulated environments
- *Mission Plan:* the robot's designated starting position, goal location, and any intermediate waypoints
- *Robot Configuration:* key parameters relevant to the specific robot setup being tested

This standardized configuration enables automated scenario generation by external tools (e.g., Surrealist) without modifying the ANYmal code base. As shown in Figure 3, the interface bridges internal software and simulation dependencies, translating configuration parameters into commands for the ANYmal stack and Gazebo. To ensure portability and integration, the entire system—including the simulator, ANYmal software, and interface—is dockerized.

B. Aerialist Adaptations (ANY Aerialist)

Originally developed for UAV testing, Aerialist [17] was closely tied to PX4's features and simulation. Yet, its core idea—abstracting dependencies for black-box testing—is widely applicable. To extend Aerialist to other use cases like ANYmal, it was refactored to decouple the generic test definition, orchestration, and analysis from the use-case-specific test execution. This transformation evolved Aerialist from a PX4/UAV-specific tool into an extensible test bench for robotic navigation systems. Key modifications include:

- *Refactoring for Extensibility:* The core Aerialist code was refactored for better modularity by introducing abstract interfaces for robot- and simulator-specific functions.
- *External Test Execution:* Aerialist was updated to enable external test execution through the ANYmal's system test interface (previously limited to PX4 commands).

[2]https://github.com/skhatiri/Surrealist
[3]https://github.com/skhatiri/Aerialist

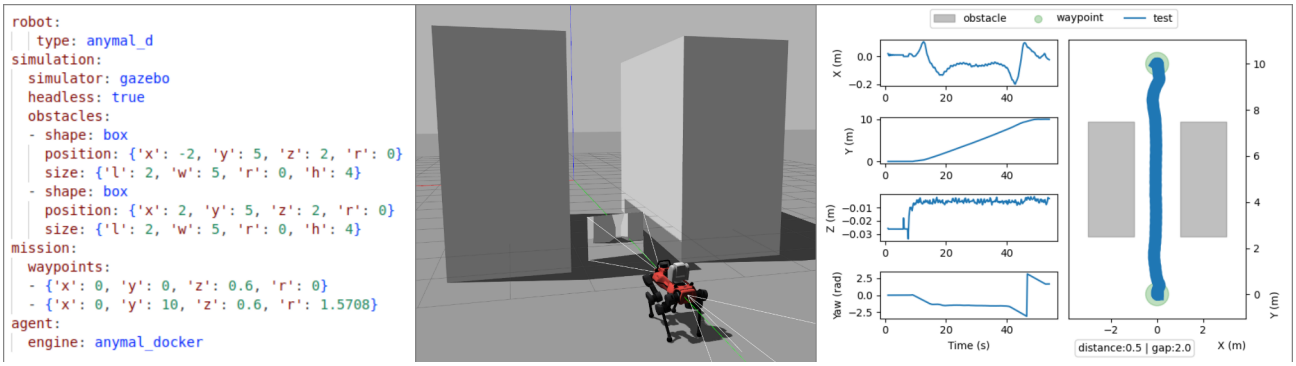

Fig. 4: Sample ANYmal test case: description (left), simulation environment (middle), robot trajectory (right)

- *ROS Bag Support:* Aerialist was modified to read ROS bag files—the standard format in ROS (Robot Operating System) used by ANYmal—instead of PX4 ulog files.
- *Failure Categories:* We automatically distinguish common categories of test outcome: *Success* (the robot reaches the navigation goal within a tolerated position error, in the given time, and without colliding against obstacles), *Safety-Stop* (a safety function prevents the robot to move further when it gets critically close to obstacles), *Timeout* (the robot can not reach the goal in time).
- *Geometric Calculations:* Distance functions were updated to model ANYmal as a box, reflecting its true size and shape, rather than treating it as a point-mass UAV.
- *Enhanced Plotting:* Visualization improvements include thick trajectory lines showing the robot's size and data on minimum robot-to-obstacle *distance* and obstacle-to-obstacle *gap* (Figure 4, right). This tool was key for the ANYbotics team for quickly diagnosing test failures and identifying common failure patterns.

C. Surrealist Adaptations (ANY Surrealist)

While Surrealist supports two use cases, we focus only on test generation, leaving real-world test replication for future work. Since Surrealist's dependency on PX4 are abstracted via Aerialist, the above adaptations to Aerialist were enough to enable test case generation for ANYmal by the existing version of Surrealist. However, Surrealist was further enhanced for quadrupedal robot navigation, based on initial feedback from the ANYbotics team. Key adaptations include:

- *Fitness Function:* We adapted the original Surrealist fitness function, aiming to minimize the robot's minimum *distance* to obstacles over test generation iterations. According to developers, this is still a good proxy for the new context, since the likelihood of failures (i.e., collisions) increases if the robot navigates too close to obstacles.
- *Seed Solutions:* Surrealist generates test suites based on a given *seed test*. The algorithm manipulates the size and position of the obstacles in the environment to find challenging test cases based on the above fitness function. We manually crafted seed tests reflecting common and challenging industrial inspection tasks (e.g., narrow corridors, doorways, cluttered spaces), based on ANYbotics' input.
- *Waypoint Mutation:* For certain test scenarios (e.g., robot passing a standard doorway), it is desirable to manipulate mission waypoints (i.e., the starting position) instead of the obstacles to find corner cases, as the robot may detect the same obstacles from different viewpoints. We implemented new mutations to move the waypoints in the environment, addressing such requests.
- *Test Suite Re-execution:* Added support for rerunning existing test scenarios to enable regression testing across navigation algorithm versions.
- *Performance Metrics:* Extended logging to capture performance and quadrupedal-specific metrics per test, including minimum obstacle distance, gap between obstacles, path lengths, deviation, and test duration.
- *Metrics Aggregation:* Aggregate logs summarize test outcomes (success, safety-stop, timeout) with descriptive statistics (min, max, average) of the above performance metrics.

D. Development Process and Integration Workflow

We forked the public Surrealist and Aerialist repositories for ANYmal-specific adaptations, refactoring the core frameworks to improve modularity while keeping the forks synchronized for easy maintenance. This design proved effective, as the integration required minimal code: Surrealist was extended with just three new classes extended from generic base classes (∼350 LOC), and Aerialist with two (∼250 LOC). These minimal additions underscore the framework's modularity and facilitate extension to other robotic use cases, following the detailed guidelines documented in the public repositories.

The full integration—including developing the ANYmal interface, refactoring the framework, and testing—took three months, with the first author dedicating roughly 80% of his time. Support from the ANYbotics team, facilitated through bi-weekly meetings and ongoing feedback, was essential to the success of this process.

IV. Research Methodology

Our empirical study evaluates our integrated simulation-based test generation framework to enhance quadrupedal robot navigation robustness, focusing on obstacle avoidance. It has two phases: (1) adapting and validating our framework on

ANYbotics' navigation software with two experimental obstacle avoidance algorithms, and (2) integrating it into ANYbotics' workflow. We use quantitative metrics and qualitative feedback, guided by four research questions:

- *RQ_1 [Development Process]:* How does the integrated framework improve the development workflow and testing practices at ANYbotics?
- *RQ_2 [Failure Detection]:* How effective is Surrealist at generating test cases that reveal failures in ANYmal's obstacle avoidance?
- *RQ_3 [Improvement Assessment]:* Can Surrealist track and quantify performance improvement across navigation software versions?
- *RQ_4 [System Verification]:* How does the integrated framework enhance the verification process for the ANYmal navigation, ensuring robustness and reliability?

A. Pilot

In the pilot phase, we evaluated the integrated framework using two experimental obstacle avoidance algorithms from ANYbotics, treated as black-box test subjects: *Exp-Nav-A* and *Exp-Nav-B*. This demonstrated our approach without accessing their proprietary navigation stacks. The experiments were:

1) *Scenario Definition (RQ_1):* We created five seed scenarios based on real-world use cases like narrow corridors and cluttered spaces.
2) *Test Generation (RQ_2):* We used Surrealist to generate a test suite (TS-Exp-A) from the seed tests for the first algorithm (Exp-Nav-A) and analyzed its performance.
3) *Algorithm Comparison (RQ_3):* We executed the same test suite (TS-Exp-A) on a newer algorithm (Exp-Nav-B) to compare their performance.
4) *Test Suite Comparison (RQ_2/RQ_3):* We generated a new test suite (TS-Exp-B) tailored to Exp-Nav-B to assess how algorithm differences affect test generation.

During the three-month pilot, the team included the first author as lead developer and three ANYbotics supervisors (navigation and test engineers). Iterative feedback enabled continuous improvements. We evaluated the obstacle avoidance performance in the generated scenarios using the metrics introduced in Section III-C including the Mission Success Rate and Safety-Stop Rate. The test generation process was conducted by the first author on a development laptop equipped with *Intel Core Ultra 9 185H* CPU, *NVIDIA RTX 2000 Ada* GPU, and *64 GB* of RAM. The first author analyzed each algorithm's overall performance, identified weaknesses and behavioral differences, and reported findings to the development team. The pilot phase primarily addresses $RQ_{1,2}$ [Development Process, Failure Detection], while providing preliminary insights for RQ_3 [Improvement Assessment].

B. Deployment

Following the successful pilot, we deployed the framework at ANYbotics, where the navigation team integrated it to evaluate in-house obstacle avoidance algorithms. A senior engineer, not involved in the pilot, led its use for failure identification, analysis, performance comparison, and improvement guidance. The first author provided technical support, while four other navigation team members participated in the discussions. In the first six months post-pilot, we tested multiple versions of the proprietary ANYmal navigation stack, including the current release referred to as *ANY-Nav-A*, and four internal candidates for the next release, referred to as *ANY-Nav-B_{1-4}*:

1) *Test Generation (RQ_2):* The ANYbotics team used Surrealist to generate a test suite (TS-ANY-A) for ANY-Nav-A, based on a selection of 3 out of 5 pilot study seed scenarios. They assessed its performance and identified the deficiencies to improve in the next version.
2) *Algorithm Comparison (RQ_3):* During their development workflow for the next version, they used the tests in TS-ANY-A as a benchmark. They executed the same tests with ANY-Nav-B_{1-4} as test subjects to assess performance improvements and behavioral differences.
3) *Targeted Test Generation (RQ_2 & RQ_4):* During ANY-Nav-A field tests, engineers found two new failure types. The development team asked the first author to create Surrealist test suites targeting these failures for simulation-based diagnosis and resolution.

Since the quantitative results from the six-month deployment are confidential (including the generated test suites, the identified failure cases, and the performance of proprietary algorithms), we evaluated the framework's industrial impact primarily through a formal questionnaire. To provide context, participants from various teams (navigation, locomotion, perception, and verification) first watched a 14-minute video demonstrating the project's workflow. After establishing participant context with demographics, the survey assessed the usability of the end-to-end workflow, from installation and test definition with Aerialist to automated generation with Surrealist. It then evaluated the framework's effectiveness based on the test suite quality, benchmarking performance, and targeted debugging. Finally, the survey assessed the system's overall impact and future directions. While some details are confidential, general findings from this survey are discussed in Section V, contributing to RQ_{1-4} [Development Process, Failure Detection, Improvement Assessment, System Verification] with an emphasis on real-world adoption and impact.

V. Evaluation Results

This section presents the evaluation results, organized according to the four research questions outlined in Section IV. For each question, we synthesize findings from both the pilot and deployment phases, combining quantitative metrics with qualitative feedback from the ANYbotics team. The quantitative analysis primarily uses metrics from the pilot study to objectively address $RQ_{2,3}$. The qualitative insights, drawn from iterative discussions and a formal survey with engineers, provide the industrial perspective on RQ_{1-4}.

A total of 8 engineers from ANYbotics participated in the feedback survey, representing a diverse range of relevant teams: Navigation (including 3 navigation engineers, a team lead, and a product manager), Locomotion (1), Perception (1),

TABLE I: Performance comparison of the pilot test subjects.

Test Subject	Test Suite (#)	$boxes_1$		$boxes_2$		corridor		cylinders		L-corridor		overall	
		Succ.	S-Stop	Succ.	S-Stop	Succ.	S-Stop	Succ.	S-Stop	Succ.	S-Stop	Succ.	S-Stop
Exp-Nav-A	TS-Exp-A (429)	39.2%	40.5%	42.1%	44.9%	37.5%	39.8%	75.0%	23.2%	23.1%	52.9%	40.3%	42.2%
Exp-Nav-B	TS-Exp-A (429)	81.1%	5.4%	69.2%	0.0%	75.0%	12.5%	98.3%	1.8%	48.1%	16.4%	71.2%	7.7%
Exp-Nav-B	TS-Exp-B (422)	72.5%	12.2%	89.8%	8.3%	64.7%	11.8%	76.3%	22.0%	34.8%	5.6%	68.3%	11.1%

and Verification (1). Participants were highly experienced and educated; 7 participants have more than three years of professional robotics experience, and all hold a master's degree (5) or PhD (3) in either Robotics or Electrical Engineering. Their self-reported expertise is strong, with most participants rating themselves as 'proficient' or higher in robotic navigation, general robotic testing, and simulation-based testing, including a test automation expert. Regarding the integrated framework itself, 3 participants had direct 'hands-on experience' or were a 'regular user', while the remaining 5 were 'conceptually familiar' through presentations and discussions.

A. RQ_1 [Development Process]

1) Pilot: The pilot phase used an iterative process guided by continuous feedback from the ANYbotics team via informal discussions and demos. This collaboration was key to improving usability and practical value. Many features in Sections III-B, III-C were added based on their input: improved plotting, clearer test outcome distinctions, and scenario selection in Aerialist, as well as test suite re-runs, waypoint mutation, and detailed metric collection in Surrealist.

2) Deployment: In the deployment phase, feedback from the formal questionnaire (Likert scale: 1 = very low, 5 = very high) confirmed the system's overall usability and its positive impact on development workflows. Setup documentation received a high average rating of 4.1, although the installation process was perceived as moderately complex (average = 3). The primary challenge was coordinating the Docker containers, as it *"still requires a few manual steps to get it working"*, suggesting the need for *"a bash/python script that installs everything."*

The YAML-based test definition with *Aerialist* was rated highly for usability (average = 4.5), with 75% of participants (6 out of 8) finding it *'easier'* or *'much easier'* to use than their previous approaches. Test execution was similarly well received (average = 4.6). The built-in automated visualizations were also praised (average = 4.3), with around 60% of participants rating them as *'better'* or *'much better'* than alternative tools. Suggestions for enhancement included more interactive and detailed features to align with tools like Foxglove[4]: *"adding the speed over time (change in color?) [...] grid on the plots [...] and an option for interactive plot would be super cool"*.

Regarding automated test generation with *Surrealist*, defining a seed scenario was rated as moderately easy (average = 3.5), while all participants considered generating a test suite from it highly intuitive (average 4.1). In terms of efficiency, 37% of participants reported a *'significant improvement'* over existing methods, while 50% considered it *'comparable, with the main benefit being automation rather than speed'*.

[4]https://foxglove.dev/

Finding 1: The integrated framework streamlined testing workflows, with Aerialist rated highly for usability and visualization (75% preferred it over prior tools) and Surrealist praised for intuitive test generation. Developers reported reduced manual effort and a faster and more productive test creation.

B. RQ_2 [Failure Detection]

1) Pilot Study: During the pilot, we evaluated Surrealist's ability to generate failure-inducing scenarios for two experimental navigation algorithms developed at ANYbotics. We measured each algorithm's *Success Rate*—the percentage of tests where the robot reached its goal safely—and *Safety Stop (S-Stop) Rate*—the percentage of failures caused by the robot's collision safety layer, which halts the robot when its current velocity is predicted to cause a collision with nearby obstacles.

We first selected five representative navigation scenarios (seeds) reflecting common and challenging industrial environments. A simple 10m task from $(x, y) = (0, 0)$ to $(0, 10)$ was manually designed by the first author, featuring five obstacle configurations. Symmetric setups included the *$boxes_1$* scenario with two 1×1m boxes (Figure 1-left), two 1m-diameter *cylinders*, and a narrow *corridor* formed by two 5m walls (seed for Figure 5-left), all spaced 2m apart. Asymmetric setups included *$boxes_2$* with two offset boxes, and an *L-corridor* (seed for Figure 5-right). We used these five seeds to generate the test suite *TS-Exp-A*, tailored for the *Exp-Nav-A* algorithm.

As shown in Table I (first row), the generated suite, which included 429 test cases in total, effectively exposed failures in Exp-Nav-A, which had a low overall success rate of 40.3% and S-Stops accounted for 42.2% of outcomes. Figure 5 (top) illustrates test cases and robot behavior in two scenarios. The algorithm performed well with wider obstacle gaps but struggled significantly when gaps were $< 1.5\,m$, either behaving risky and leading to a safety-stop, or behaving too cautions and not entering wide enough gaps, leading to a timeout while looking for other ways around. The *L-corridor* scenario was especially difficult, with only a 23.1% success rate and 52.9% S-Stops, highlighting issues navigating tight spaces safely.

2) Deployment: While specific quantitative performance data from the deployment phase remains confidential, feedback from the formal questionnaire confirmed Surrealist's effectiveness in uncovering failures. Participants rated (with Likert scale: 1 = very low, 5 = very high) the generated test cases

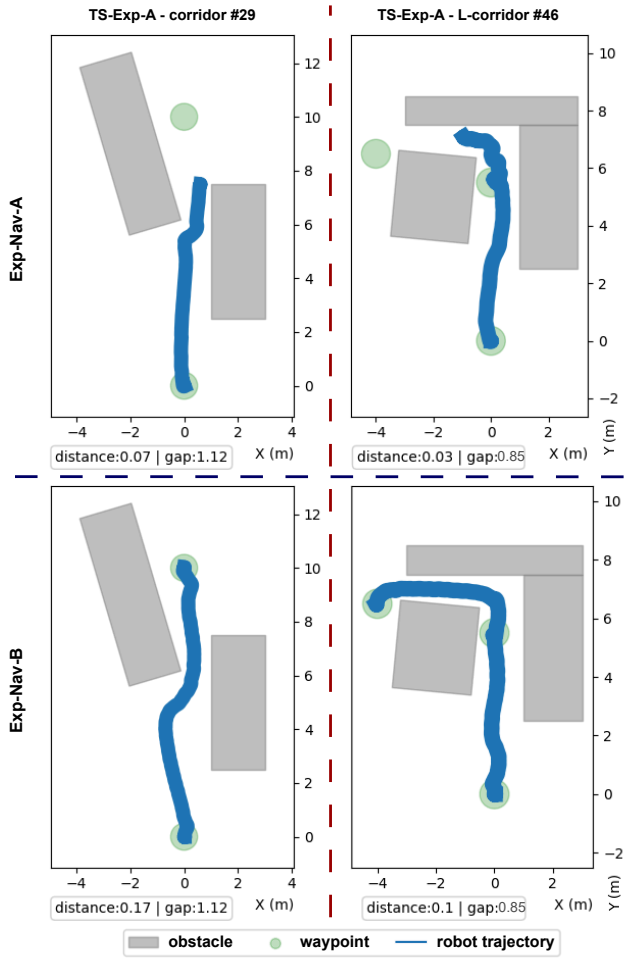

Fig. 5: Robot trajectory comparison with different obstacle avoidance algorithms in the same test environments

as highly realistic and relevant, with an average score of 4.3. Most participants reported marginal (50%) or no concern (50%) about the sim-to-real gap, with an average rating of 4.5 (where 5 means not at all concerned). The scenarios were perceived as challenging, receiving an average rating of 3.6. Notably, 50% of participants found these scenarios more challenging than their typical manual test cases. Participants found the scenarios effective in exposing realistic failures, noting they matched those seen on real robots: *"I never saw a failure case that we did not yet see with Surrealist"*. Scenarios with "narrowing gaps" (similar to the corridors in Fig. 5) were found specially realistic and representative of challenging industrial environments. The diversity of generated tests was appreciated, even if some were oversimplified, as the goal is to *"break the system in test"*. Concerns about the sim-to-real gap centered on perception noise, dynamic elements, and oversimplified geometry: *"obstacles in industrial settings are rarely as uniform as the cubes or cylinders used in testing"*. While the gap between simulation and reality was seen as manageable, especially for navigation, one participant still

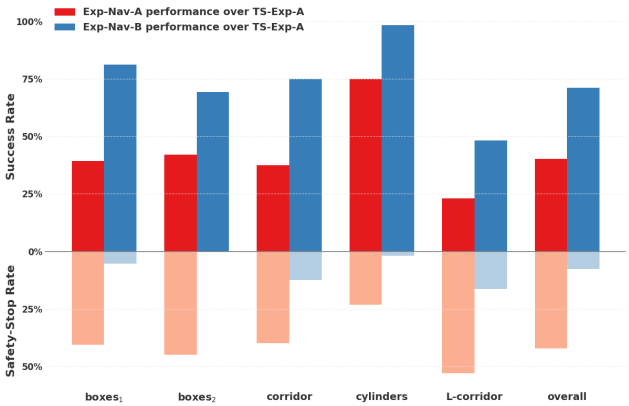

Fig. 6: Performance comparison of the pilot test subjects for TS-Exp-A in a two-sided bar chart. The second iteration of the obstacle avoidance algorithm showed a clear improvement (significantly higher success rate and lower safety-stop rate) over the initial prototype.

recommended real-world verification: *"I would still retest... to verify the behavior in the real world"*.

Importantly, all 8 participants rated the framework as either Effective (62%) or Very Effective (38%) at identifying previously unknown failures and corner cases. This was reinforced by qualitative feedback; for example, one engineer commented: "Unknown edge cases. This is were surrealist shines the most!". Moreover, the use case of generating a test suite based on a known real-world failure was highly valued. Engineers gave this approach strong ratings for 'relevance' (average 4.1) and 'solution validation' (average 4.4), though ratings were more moderate for 'root cause analysis' (average 3.3). The effort to convert a real-world issue into a simulation seed was rated as 'high' by 37% of participants, highlighting a need for future automation, e.g., by *"[automatically] creating test scenarios from collected point clouds/depth data"*.

Finding 2: Surrealist's search-based approach effectively generated challenging scenarios that revealed critical failures. In the pilot, it reduced a preliminary navigation algorithm's success rate to 40.3%. All engineers rated it as effective for uncovering previously unknown failures and corner cases during deployment. Scenarios were considered both realistic (avg. 4.5) and challenging (avg. 3.6). The diversity of generated tests was valued positively for stress-testing the system.

C. RQ_3 [Improvement Assessment]

1) Pilot Study: A key feature we implemented during the pilot enabled the re-running of generated test suites, giving ANYbotics a repeatable, quantitative method to compare algorithms. To evaluate system improvements, we tested two experimental obstacle avoidance algorithms, Exp-Nav-A and Exp-Nav-B, on the same challenging test suite, TS-Exp-A, and measured their success and S-Stop rates. From Table I (rows 1-2), Exp-Nav-B outperformed Exp-Nav-A with a success rate of 71.2% vs. 40.3%, and a significantly lower S-Stop rate of

7.7% vs. 42.2%. This is even more evident in the comparison chart in Figure 6, where Exp-Nav-B consistently shows higher success and lower s-stop rate in all test scenarios.

The qualitative nature of this improvement is visually evident in the test examples shown in Figure 5. Across different scenarios, Exp-Nav-B consistently executes smoother, more confident paths with safer clearances from obstacles. For example, in the L-corridor #46 scenario, Exp-Nav-B performs a wide, clean turn (distance to closest obstacle: 0.1m), whereas Exp-Nav-A takes a very tight, hesitant turn that brings it dangerously close to the corner (distance to closest obstacle: 0.03m) which causes the robot's collision safety layer to halt the robot. This demonstrates that the framework not only quantifies success or failure but also provides visual evidence to assess the robot's behavior, making it a relevant tool for validating and comparing improvements.

We then examined the framework's capability to adapt its test generation process in order to expose the specific weaknesses of the more robust Exp-Nav-B algorithm. To this end, we generated a new test suite (TS-Exp-B) using the same seed scenarios used before, but this time optimized for the Exp-Nav-B algorithm. As shown in Table I (rows 2–3), the performance of Exp-Nav-B declined when evaluated on its own custom-generated test suite (422 test cases). Specifically, the overall success rate dropped from 71.2% to 68.3%, while the rate of safety stops increased from 7.7% to 11.1%. This pattern was consistent across most test scenarios, with the notable exception of $\texttt{boxes}_2$, where the success rate actually improved. However, even in that scenario, Surrealist identified new safety-critical configurations, resulting in an increase in the safety stop rate from 0% to 8.3%. These results demonstrate that our search-based framework is capable of effectively uncovering and targeting the distinct vulnerabilities of each system under test, producing a uniquely challenging test suite tailored to each individual implementation.

2) Deployment: The value of the framework's comparative assessment capability was strongly confirmed during the industrial deployment, where the ANYbotics team actively used the test suite re-execution feature to benchmark new implementations against the current release, and analyze their pros and cons to inform enhancement decisions. Feedback from the formal questionnaire highlighted the feature's high utility. All participants agreed that Surrealist can be used to compare different algorithms, rating its usefulness as higher than their previous manual comparison methods, with an average score of 4.6. Furthermore, the performance metrics provided by the system were considered 'very relevant' for this purpose by most participants (7 out of 8). Qualitative feedback provided suggestions for future improvements. On the analysis side, engineers suggested the development of a dedicated tool for simultaneous visualization of two test runs to make algorithm comparison less time-consuming, and the addition of confidence intervals to success rates to enhance statistical rigor. Furthermore, participants suggested new performance metrics beyond mission success, with one engineer noting it *"would be nice to have some 'efficiency' metric that measured how close was the robot to the 'optimal'/'shortest' path to the goal and how fast it moved along it."*

Finding 3: The framework enables repeatable, quantitative comparisons of navigation algorithms across software versions. It distinguished two preliminary approaches (71.2% vs. 40.3% success rate, 7.7% vs. 42.2% safety stop rate), with qualitative visual evidence confirming behavioral improvements. During deployment, engineers rated the benchmarking feature highly (on average 4.6 out of 5), considering it a significant upgrade over previous methods and used to guide their enhancement efforts.

D. RQ_4 [System Verification]

1) Pilot: The pilot study showed that Surrealist can aid system verification by generating challenging scenarios and exposing failures. Its ability to reveal issues—even with a simplified navigation stack—highlights its potential for uncovering weaknesses in more complex systems.

2) Deployment: Survey results from the deployment phase confirm that the framework adds significant value to system verification. All participants *'agreed'* or *'strongly agreed'* that it increased their confidence in the system's robustness, and a strong majority (7/8) said it improved their team's verification ability. This was attributed to its effectiveness in specific Verification and Validation (V&V) tasks: All participants rated the system as 'better' or 'much better' than manual design for finding difficult-to-predict corner cases, and its ability to re-run test suites for regression testing was considered 'extremely valuable' by 86% of the engineers (6 out of 7 responses). The generated scenarios and visualizations were also found to be very helpful in debugging the root cause of failures (average rating of 3.9). Looking at the broader workflow, a majority of participants (71%) found the approach 'very valuable' for prioritizing physical, real-world tests, and a majority (6 out of 8) see high or some potential for using the generated reports as formal V&V evidence. The framework's critical role was summarized by one engineer who stated they "would not release [new navigation algorithms] if they were not tested with surrealist", positioning the tool as an essential pre-release validation gate. Complimentary, another engineer stated that "a further validation phase in real world would be needed to cover unforeseen real world interactions/onboard resources limitations or judging qualitatively the behaviors."

Finding 4: The framework enhances system verification and developer confidence by systematically uncovering difficult-to-predict corner cases and enabling efficient regression testing. Participants rated it superior to manual test design, found its debugging support highly useful, and called it an essential pre-release validation step.

VI. Discussion

Industrial Impact & Adoption. Our research demonstrates the successful adaptation and industrial integration of the

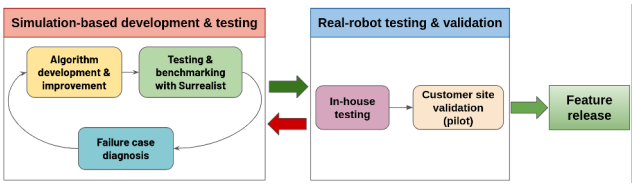

Fig. 7: ANYbotics' development workflow for releasing new navigation algorithms includes both simulation and real-robot testing stages. Our approach integrates directly into the simulation stage, allowing for early detection of critical failures and deficiencies before they reach real-robot tests.

Surrealist and Aerialist frameworks into the development workflow of the ANYmal quadrupedal robot at ANYbotics. Through a two-phase evaluation, we confirmed that our approach (i) enhances the development process (RQ_1), (ii) effectively uncovers critical navigation failures (RQ_2), (iii) provides objective benchmarks for system improvement (RQ_3), and (iv) significantly strengthens the overall verification pipeline (RQ_4). The successful deployment of our framework for testing a proprietary navigation stack marks a key milestone, transitioning our search-based testing methodology from a research prototype to a solution demonstrated in its intended operational environment—the ANYbotics development workflow (Technology Readiness Level 7 [56]). The framework's adoption and impact are reflected in the survey results: 4 out of 8 participants identified themselves as 'current users'. The impact on the team's workflow was rated as 'very positive' by 6 out of 8 engineers. The majority reported they will use the tool "occasionally for specific tasks" (63%) and on a monthly or weekly basis (63%), a strong indication of sustainable integration into their development processes. As one engineer noted, the ability of the tool to create *"test cases that reflect challenges observed later in the real world [...] built up trust quite a bit"*. Furthermore, the ANYbotics Locomotion team is already making efforts to "adopt a similar search-based testing approach", demonstrating how the project's impact extends beyond the initial navigation context and is shaping the company's long-term testing strategy.

Broader Applicability Across Domains & Development Stages. The framework—combining automated test generation with a modular abstraction interface—has matured to a point where it can be applied beyond UAVs and quadrupedal robots. It is designed for easy extension to other domains, such as wheeled robots and manipulators, as long as an appropriate simulation interface is available. Complementary, this study offers key insights into the practical applications of automated test generation in industrial robotics. The survey provided a clear ranking of development activities, where the tool provides the most value. *Regression Testing* emerged as the top use case selected by all participants (8/8), followed by *MLOps* (6/8), highlighting the need for further automation and optimizations for a fast-paced ML development cycle. Other highly valued applications included *Prototyping & Early Feature Development* (5/8), *Exploratory Testing* (5/8), *Failure Replication & Debugging* (4/8) and *Release Validation* (4/8).

This versatility allows the framework to be deeply integrated into the ANYbotics development workflow, as illustrated in Fig. 7. ANYbotics releases new features following a two-stage process: development and testing in simulation, followed by deployment on real hardware—both at internal testing facilities and pilot customer sites. Our framework fits seamlessly into the simulation phase, enabling more thorough and efficient testing and benchmarking early in the pipeline. This early-stage integration supports the timely identification of algorithmic weaknesses, speeding up the entire development process by minimizing issues that would otherwise surface during the more costly and time-consuming real-world testing stages. Ultimately, this leads to more robust and reliable releases.

Feedback from Study Participants. While the framework proved highly effective for its intended purpose, the industrial deployment and the detailed feedback from the survey also highlighted its current limitations as well as a clear path for future evolution. The suggestions centered on three key areas:

1) *Further bridge the sim-to-real gap*: add more realistic sensor noise models, generate complex 3D terrains (such as stairs and ramps), incorporate a broader range of irregular obstacles (such as those found in industrial settings), and add support for dynamic obstacles.
2) *Streamline the debugging process*: introduce a "real-to-sim" capability to auto-generate seeds from real-world failure data. While this is a core feature of the original Surrealist for UAVs, feedback reaffirms its importance in reducing the high manual effort for seed creation. Also, a dedicated visualization tool to compare algorithm runs side-by-side and improved statistical metrics were suggested to enhance performance benchmarks.
3) *Extension*: the framework's success in navigation has sparked interest in expanding its scope, e.g., handling more complex challenges—such as navigation without predefined waypoints—and extending it to new domains like locomotion testing.

A. Takeaways for Researchers and Robotic Practitioners

- *Integrate with a Non-Invasive Abstraction Layer:* The success of this project relied on ANYmal's system test interface — a facade we developed to let our research prototypes (Surrealist and Aerialist) interact with ANYbotics' proprietary software stack in a black-box, non-invasive way. This was key for industrial adoption, as it avoided disrupting existing workflows. Additionally, Aerialis's generic test description (in YAML) and the framework's decoupled architecture enabled a smooth extension from UAVs to the quadrupeds.
- *Adopt a User-Driven, Iterative Development Process:* In addition to advancing core test generation algorithms, it is important for researchers to also consider the usability of testing tools and the practical relevance of their outputs to support broader industrial adoption. The pilot phase—with its close, iterative feedback loop involving *ANYbotics* engineers —was instrumental in refining the framework. Direct user feedback led to meaningful improvements in visualization, performance metrics, and essential features such as test

suite re-execution, ultimately enhancing the tool's practical utility and enabling its successful real-world deployment.

- *Address the Sim-to-Real Gap Strategically:* The survey revealed the sim-to-real gap is a relative, not absolute, barrier. It was considered manageable for navigation, where the robot is modeled accurately, the environment is abstracted through sensor preprocessing, and movements are streamlined by robust locomotion. Here, the framework successfully predicted real-world algorithmic failures, building significant trust with the engineers. The primary concerns were perception-related (e.g., lack of sensor noise, oversimplified obstacles), which reinforces a practical workflow: use simulation for broad, cost-effective discovery of high-level algorithmic flaws, and reserve targeted physical testing for scenarios where perception performance may degrade.

B. Threats to Validity

While our findings support the positive impact of the integrated framework, several threats to validity must be considered. The results are fundamentally dependent on the fidelity of the simulation environment and are subject to the well-known "reality gap": behaviors observed in simulation may not perfectly translate to the physical world. Our test generation process is rooted in a limited set of manually designed seed scenarios and is guided by a fitness function focused primarily on obstacle proximity. These choices, while effective, may not capture all relevant edge cases or dimensions of scenario difficulty. Similarly, the performance metrics used do not encompass all aspects of navigation quality, such as energy efficiency or motion smoothness. Finally, while the evaluation was conducted on an industrial-grade platform, the findings are specific to the ANYmal robot and the organizational context of a single company, and the qualitative insights are derived from a small group of eight engineers.

VII. Related Works

Testing of robotic systems—especially for autonomous navigation—remains difficult due to system complexity and unpredictable environments [20]. Designing realistic test environments and oracles is notoriously challenging [20], [57], limiting the adoption of simulation-based testing despite its benefits [28]. The gap between simulated and real-world performance further complicates validation [28]. Our work addresses these challenges by extending our prior work on Surrealist [9]—a search-based approach for generating challenging test variants for UAVs—to quadrupedal robots and integrating it into an industrial workflow.

A significant body of research focuses on generating diverse environments to improve test coverage. Procedural content generation, as explored by Sotiropoulos et al. [58], uses randomization to create open-space worlds to find navigation bugs, with a focus on assessing the difficulty of these worlds [59]. To provide more structure, Parra et al. [60] introduced FloorPlan DSL for defining indoor test environments and Variation DSL for structured variability, such as sampling obstacle sizes from a normal distribution. Similarly, tools like *Local Planner Bench* [61] support randomized environments to benchmark local obstacle avoidance algorithms. While these methods excel at creating a wide variety of test scenarios, they are generally undirected. In contrast, our approach uses a search-based algorithm to systematically and purposefully evolve scenarios to find failure-inducing configurations.

Another line of research focuses specifically on automated techniques to find failures. Fuzzing methods, such as PHYS-FUZZ by Woodlief et al. [29], vary environmental parameters and robot poses to uncover crashes in mobile robots. This is conceptually similar to approaches in autonomous driving, where simulation is used to generate failure scenarios by altering the environment, traffic behavior, and sensor inputs [14], [25], [62], [63].

Search-based techniques offer a more guided approach to failure discovery. For instance, Humeniuk et al. [64] proposed MARTENS, a search-based method for testing the DL vision models used in autonomous manipulators. Our work also falls into this category, but while prior approaches often focus on specific input space exploration (fuzzing) or perception modules (MARTENS), our work targets the full-system behavior of an industrial quadrupedal robot. We use search to optimize for challenging dynamics in obstacle avoidance, offering a more comprehensive, system-level testing strategy. Our successful integration and evaluation on the ANYmal platform further highlight the real-world applicability of this approach for improving the robustness of complex legged robots.

VIII. Conclusion and Future Work

This paper successfully demonstrated how an academic, search-based testing framework can be adapted and integrated into a complex industrial robotics workflow. Our two-phased evaluation at ANYbotics confirmed the framework's value: it significantly streamlined the testing process, uncovered critical failures, and provided an objective means to benchmark performance improvements. This work serves as a case study for bridging the research-to-practice gap, delivering a mature, industrially-validated tool that enhances the robustness of the ANYmal robot and accelerates the overall development cycle.

Future work will focus on expanding the framework's capabilities based on the received feedback. Our primary research directions include enhancing scenario realism by incorporating more complex environmental features and sensor models, replicating real-world tests in simulation, and broadening the framework's applicability to new robotic domains.

Acknowledgments

We thank the Horizon and SERI for supporting the Inno-Guard project (Marie Skłodowska-Curie DN, HORIZON-MSCA-2023-DN), the SNSF for the "SwarmOps" project (No. 200021_219732), and the Hasler Foundation for the projects "Aerialist" (No. 200021_219732) and "Safe2Fly" (No. 2025-02-27-311). We sincerely thank ANYbotics AG for their full support of this study, with special gratitude to Gabriel Hottiger, Rene Hölbling, and Yi Hao Ng for their supervision, and all survey participants.

REFERENCES

[1] F. Ingrand, "Recent trends in formal validation and verification of autonomous robots software," in *3rd IEEE International Conference on Robotic Computing, IRC 2019, Naples, Italy, February 25-27, 2019*, 2019, pp. 321–328.

[2] S. Halder and K. Afsari, "Robots in inspection and monitoring of buildings and infrastructure: A systematic review," *Applied Sciences*, vol. 13, no. 4, 2023. [Online]. Available: https://www.mdpi.com/2076-3417/13/4/2304

[3] H. Huang, A. V. Savkin, M. Ding, and C. Huang, "Mobile robots in wireless sensor networks: A survey on tasks," *Computer Networks*, vol. 148, pp. 1–19, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S138912861830255X

[4] L. Antonyshyn, J. Silveira, S. Givigi, and J. Marshall, "Multiple mobile robot task and motion planning: A survey," *ACM Comput. Surv.*, vol. 55, no. 10, Feb. 2023. [Online]. Available: https://doi.org/10.1145/3564696

[5] C. Torras, "Ethics of social robotics: Individual and societal concerns and opportunities," *Annu. Rev. Control. Robotics Auton. Syst.*, vol. 7, no. 1, 2024. [Online]. Available: https://doi.org/10.1146/annurev-control-062023-082238

[6] G. A. Kebede, A. A. Gelaw, H. Andualem, and A. T. Hailu, "Review of the characteristics of mobile robots for health care application," *Int. J. Intell. Robotics Appl.*, vol. 8, no. 2, pp. 480–502, 2024. [Online]. Available: https://doi.org/10.1007/s41315-024-00324-3

[7] C. Gehring, P. Fankhauser, L. Isler, R. Diethelm, S. Bachmann, M. Potz, L. Gerstenberg, and M. Hutter, "ANYmal in the Field: Solving Industrial Inspection of an Offshore HVDC Platform with a Quadrupedal Robot," in *Springer Proceedings in Advanced Robotics*. Springer Science and Business Media B.V., 2021, vol. 16, pp. 247–260.

[8] J. Guiochet, M. Machin, and H. Waeselynck, "Safety-critical advanced robots: A survey," *Robotics and Autonomous Systems*, vol. 94, pp. 43–52, 2017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0921889016300768

[9] S. Khatiri, S. Panichella, and P. Tonella, "Simulation-based test case generation for unmanned aerial vehicles in the neighborhood of real flights," in *16th IEEE International Conference on Software Testing, Verification and Validation (ICST)*, 2023.

[10] S. Nahavandi, R. Alizadehsani, D. Nahavandi, S. Mohamed, N. Mohajer, M. Rokonuzzaman, and I. Hossain, "A comprehensive review on autonomous navigation," *ACM Comput. Surv.*, vol. 57, no. 9, May 2025. [Online]. Available: https://doi.org/10.1145/3727642

[11] J. Borenstein and Y. Koren, "Real-time obstacle avoidance for fast mobile robots," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 19, no. 5, pp. 1179–1187, 1989.

[12] S. Robla-Gómez, V. M. Becerra, J. R. Llata, E. González-Sarabia, C. Torre-Ferrero, and J. Pérez-Oria, "Working together: A review on safe human-robot collaboration in industrial environments," *IEEE Access*, vol. 5, pp. 26 754–26 773, 2017.

[13] S. Frey, A. Rashid, P. Anthonysamy, M. Pinto-Albuquerque, and S. A. Naqvi, "The good, the bad and the ugly: A study of security decisions in a cyber-physical systems game," *IEEE Trans. Software Eng.*, vol. 45, no. 5, pp. 521–536, 2019.

[14] C. Birchler, T. K. Mohammed, P. Rani, T. Nechita, T. Kehrer, and S. Panichella, "How does simulation-based testing for self-driving cars match human perception?" in *ACM International Conference on the Foundations of Software Engineering*, 2024.

[15] A. Di Sorbo, F. Zampetti, A. Visaggio, M. Di Penta, and S. Panichella, "Automated identification and qualitative characterization of safety concerns reported in uav software platforms," *ACM Trans. Softw. Eng. Methodol.*, vol. 32, no. 3, apr 2023. [Online]. Available: https://doi.org/10.1145/3564821

[16] F. Zampetti, R. Kapur, M. D. Penta, and S. Panichella, "An empirical characterization of software bugs in open-source cyber-physical systems," *J. Syst. Softw.*, vol. 192, p. 111425, 2022. [Online]. Available: https://doi.org/10.1016/j.jss.2022.111425

[17] S. Khatiri, S. Panichella, and P. Tonella, "Simulation-based testing of unmanned aerial vehicles with aerialist," in *Proceedings of the 2024 IEEE/ACM 46th International Conference on Software Engineering: Companion Proceedings*, 2024, pp. 134–138.

[18] C. Birchler, S. Khatiri, P. Rani, T. Kehrer, and S. Panichella, "A roadmap for simulation-based testing of autonomous cyber-physical systems: Challenges and future direction," New York, NY, USA, 2025. [Online]. Available: https://doi.org/10.1145/3711906

[19] C. Birchler, C. Rohrbach, H. Kim, A. Gambi, T. Liu, J. Horneber, T. Kehrer, and S. Panichella, "TEASER: simulation-based CAN bus regression testing for self-driving cars software," in *38th IEEE/ACM International Conference on Automated Software Engineering, ASE 2023, Luxembourg, September 11-15, 2023*. IEEE, 2023, pp. 2058–2061. [Online]. Available: https://doi.org/10.1109/ASE56229.2023.00154

[20] A. Afzal, C. Le Goues, M. Hilton, and C. S. Timperley, "A study on challenges of testing robotic systems," in *International Conference on Software Testing, Validation and Verification*. IEEE, 2020, pp. 96–107.

[21] H. Araujo, M. R. Mousavi, and M. Varshosaz, "Testing, validation, and verification of robotic and autonomous systems: A systematic review," *ACM Trans. Softw. Eng. Methodol.*, vol. 32, no. 2, Mar. 2023. [Online]. Available: https://doi.org/10.1145/3542945

[22] A. Afzal, C. L. Goues, M. Hilton, and C. S. Timperley, "A study on challenges of testing robotic systems," in *2020 IEEE 13th International Conference on Software Testing, Validation and Verification (ICST)*, 2020, pp. 96–107.

[23] T. Zohdinasab, V. Riccio, A. Gambi, and P. Tonella, "Efficient and effective feature space exploration for testing deep learning systems," *ACM Trans. Softw. Eng. Methodol.*, vol. 32, no. 2, pp. 49:1–49:38, 2023. [Online]. Available: https://doi.org/10.1145/3544792

[24] C. Birchler, S. Khatiri, P. Derakhshanfar, S. Panichella, and A. Panichella, "Single and multi-objective test cases prioritization for self-driving cars in virtual environments," *ACM Trans. Softw. Eng. Methodol.*, vol. 32, no. 2, pp. 28:1–28:30, 2023. [Online]. Available: https://doi.org/10.1145/3533818

[25] C. Birchler, S. Khatiri, B. Bosshard, A. Gambi, and S. Panichella, "Machine learning-based test selection for simulation-based testing of self-driving cars software," *Empir. Softw. Eng.*, vol. 28, no. 3, p. 71, 2023. [Online]. Available: https://doi.org/10.1007/s10664-023-10286-y

[26] J. Zhou, Y. Gao, O. Johansson, B. Olofsson, and E. Frisk, "Robust predictive motion planning by learning obstacle uncertainty," *IEEE Transactions on Control Systems Technology*, vol. 33, no. 3, pp. 1006–1020, 2025.

[27] C. S. Timperley, A. Afzal, D. S. Katz, J. M. Hernandez, and C. Le Goues, "Crashing simulated planes is cheap: Can simulation detect robotics bugs early?" in *2018 IEEE 11th International Conference on Software Testing, Verification and Validation (ICST)*. IEEE, 2018, pp. 331–342.

[28] A. Afzal, D. S. Katz, C. Le Goues, and C. S. Timperley, "Simulation for robotics test automation: Developer perspectives," in *Conference on Software Testing, Verification and Validation*. IEEE, 2021, pp. 263–274.

[29] T. Woodlief, S. Elbaum, and K. Sullivan, "Fuzzing mobile robot environments for fast automated crash detection," in *International Conference on Robotics and Automation*. IEEE, 2021, pp. 5417–5423.

[30] C. Hildebrandt and S. Elbaum, "World-in-the-loop simulation for autonomous systems validation," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 10 912–10 919.

[31] S. Khatiri, F. M. Amin, S. Panichella, and P. Tonella, "When uncertainty leads to unsafety: Empirical insights into the role of uncertainty in unmanned aerial vehicle safety," *Empirical Software Engineering journal (EMSE 2025)*, 2025.

[32] S. Khatiri, P. Saurabh, T. Zimmermann, C. Munasinghe, C. Birchler, and S. Panichella, "SBFT tool competition 2024 - cps-uav test case generation track," in *IEEE/ACM International Workshop on Search-Based and Fuzz Testing, SBFT@ICSE 2024*, 2024.

[33] S. Khatiri, T. Zohdinasab, P. Saurabh, D. Humeniuk, and S. Panichella, "SBFT tool competition 2025 - cps-uav test case generation track," in *IEEE/ACM International Workshop on Search-Based and Fuzz Testing, SBFT@ICSE 2025*, 2025.

[34] ——, "ICST tool competition 2025 - uav testing track," in *IEEE/ACM International Conference on Software Testing, Verification and Validation, ICST 2025*, 2025.

[35] L. Meier, D. Honegger, and M. Pollefeys, "Px4: A node-based multithreaded open source robotics framework for deeply embedded platforms," in *international conference on robotics and automation*. IEEE, 2015, pp. 6235–6240.

[36] S. Khatiri, A. D. Sorbo, F. Zampetti, C. A. Visaggio, M. D. Penta, and S. Panichella, "Identifying safety-critical concerns in unmanned aerial vehicle software platforms with SALIENT," *SoftwareX*, vol. 27, p. 101748, 2024. [Online]. Available: https://doi.org/10.1016/j.softx.2024.101748

[37] M. Hutter, C. Gehring, A. Lauber, F. Gunther, C. D. Bellicoso, V. Tsounis, P. Fankhauser, R. Diethelm, S. Bachmann, M. Blösch

et al., "Anymal-toward legged robots for harsh environments," *Advanced Robotics*, vol. 31, no. 17, pp. 918–931, 2017.

[38] ANYbotics, "Anybotics - autonomous robotic inspection solutions," https://www.anybotics.com/, 2025, accessed: 02.20.2025.

[39] D. Hoeller, N. Rudin, D. Sako, and M. Hutter, "Anymal parkour: Learning agile navigation for quadrupedal robots," *Science Robotics*, vol. 9, no. 88, p. eadi7566, 2024.

[40] S. Ha, J. Lee, M. van de Panne, Z. Xie, W. Yu, and M. Khadiv, "Learning-based legged locomotion: State of the art and future perspectives," *The International Journal of Robotics Research*, vol. 0, no. 0, p. 02783649241312698, 0. [Online]. Available: https://doi.org/10.1177/02783649241312698

[41] S. Solmaz, P. Innerwinkler, M. Wójcik, K. Tong, E. Politi, G. Dimitrakopoulos, P. Purucker, A. Höß, B. W. Schuller, and R. John, "Robust robotic search and rescue in harsh environments: An example and open challenges," in *2024 IEEE International Symposium on Robotic and Sensors Environments (ROSE)*. IEEE, 2024, pp. 1–8.

[42] T. Miki, J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter, "Learning robust perceptive locomotion for quadrupedal robots in the wild," *Science robotics*, vol. 7, no. 62, p. eabk2822, 2022.

[43] J. Yue, "Learning locomotion for legged robots based on reinforcement learning: A survey," in *2020 International Conference on Electrical Engineering and Control Technologies (CEECT)*. IEEE, 2020, pp. 1–7.

[44] M. Liu, J. Xiao, and Z. Li, "Deployment of whole-body locomotion and manipulation algorithm based on nmpc onto unitree go2quadruped robot," in *2024 6th International Conference on Industrial Artificial Intelligence (IAI)*. IEEE, 2024, pp. 1–6.

[45] C. Yan, N. Wang, H. Gao, X. Wang, C. Tang, L. Zhou, Y. Li, and Y. Wang, "An advanced reinforcement learning control method for quadruped robots in typical urban terrains," *International Journal of Machine Learning and Cybernetics*, pp. 1–11, 2024.

[46] A. Pandey, S. Pandey, and D. Parhi, "Mobile robot navigation and obstacle avoidance techniques: A review," *Int Rob Auto J*, vol. 2, no. 3, p. 00022, 2017.

[47] T. Dudzik, M. Chignoli, G. Bledt, B. Lim, A. Miller, D. Kim, and S. Kim, "Robust autonomous navigation of a small-scale quadruped robot in real-world environments," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 3664–3671.

[48] M. Raibert, K. Blankespoor, G. Nelson, and R. Playter, "Bigdog, the rough-terrain quadruped robot," *IFAC Proceedings Volumes*, vol. 41, no. 2, pp. 10 822–10 825, 2008.

[49] Unitree Robotics, "Go2: Redefining the future of quadrupedal robots," https://www.unitree.com/go2, 2024, accessed: [19.02.2025].

[50] B. Katz, J. Di Carlo, and S. Kim, "Mini cheetah: A platform for pushing the limits of dynamic quadruped control," in *2019 international conference on robotics and automation (ICRA)*. IEEE, 2019, pp. 6295–6301.

[51] S. Gilroy, D. Lau, L. Yang, E. Izaguirre, K. Biermayer, A. Xiao, M. Sun, A. Agrawal, J. Zeng, Z. Li *et al.*, "Autonomous navigation for quadrupedal robots with optimized jumping through constrained obstacles," in *2021 IEEE 17th International Conference on Automation Science and Engineering (CASE)*. IEEE, 2021, pp. 2132–2139.

[52] B. Dai, R. Khorrambakht, P. Krishnamurthy, and F. Khorrami, "Sailing through point clouds: Safe navigation using point cloud based control barrier functions," *arXiv preprint arXiv:2403.18206*, 2024.

[53] W. Zhang, S. Xu, P. Cai, and L. Zhu, "Agile and safe trajectory planning for quadruped navigation with motion anisotropy awareness," *arXiv preprint arXiv:2403.10101*, 2024.

[54] S. Karlsson, B. Lindqvist, and G. Nikolakopulos, "Ensuring robot-human safety for the bd spot using active visual tracking and nmpc with velocity obstacles," *IEEE Access*, vol. 10, pp. 100 224–100 233, 2022.

[55] C. Gehring, P. Fankhauser, L. Isler, R. Diethelm, S. Bachmann, M. Potz, L. Gerstenberg, and M. Hutter, "Anymal in the field: Solving industrial inspection of an offshore hvdc platform with a quadrupedal robot," in *Field and Service Robotics: Results of the 12th International Conference*. Springer, 2021, pp. 247–260.

[56] J. C. Mankins *et al.*, "Technology readiness levels," 1995.

[57] M. Lindvall, A. Porter, G. Magnusson, and C. Schulze, "Metamorphic model-based testing of autonomous systems," in *International Workshop on Metamorphic Testing*. IEEE, 2017, pp. 35–41.

[58] T. Sotiropoulos, H. Waeselynck, J. Guiochet, and F. Ingrand, "Can robot navigation bugs be found in simulation? an exploratory study," in *2017 IEEE International conference on software quality, reliability and security (QRS)*. IEEE, 2017, pp. 150–159.

[59] T. Sotiropoulos, G. Guiochet, I. Ingrand, and W. Waeselynck, "Virtual worlds for testing robot navigation: a study on the difficulty level," in *2016 12th European Dependable Computing Conference (EDCC)*. IEEE, 2016, pp. 153–160.

[60] S. Parra, S. Schneider, and N. Hochgeschwender, "A thousand worlds: scenery specification and generation for simulation-based testing of mobile robot navigation stacks," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2023, pp. 5537–5544.

[61] M. Spahn, C. Salmi, and J. Alonso-Mora, "Local planner bench: Benchmarking for local motion planning," *arXiv preprint arXiv:2210.06033*, 2022.

[62] A. Gambi, M. Müller, and G. Fraser, "Automatically testing self-driving cars with search-based procedural content generation," in *ACM SIGSOFT International Symposium on Software Testing and Analysis*. ACM, 2019, pp. 318–328.

[63] A. Stocco, M. Weiss, M. Calzana, and P. Tonella, "Misbehaviour prediction for autonomous driving systems," in *International Conference on Software Engineering*, 2020, pp. 359–371.

[64] D. Humeniuk, H. Ben Braiek, T. Reid, and F. Khomh, "In-simulation testing of deep learning vision models in autonomous robotic manipulators," in *Proceedings of the 39th IEEE/ACM International Conference on Automated Software Engineering*, 2024, pp. 2187–2198.