

RoboVAST

Variation Automation and Scalable Testing for Robotic Systems

Frederik Pasch
Samuel Wiest¹

Motivation

The Testing Challenge

- Robotic systems are complex.
- Bugs and performance issues often occur in untested situations.

The Solution

- Vast Test Variation in Simulation!

“Doesn't setting up and running those tests take forever!?”

ㄟ(ツ)ㄟ

RoboVAST

Open-Source Testing Framework

- Test Definition: Floorplan and OpenSCENARIO DSL
- Test Execution: Scenario Execution
- Scalable Parallel Testing: Kubernetes

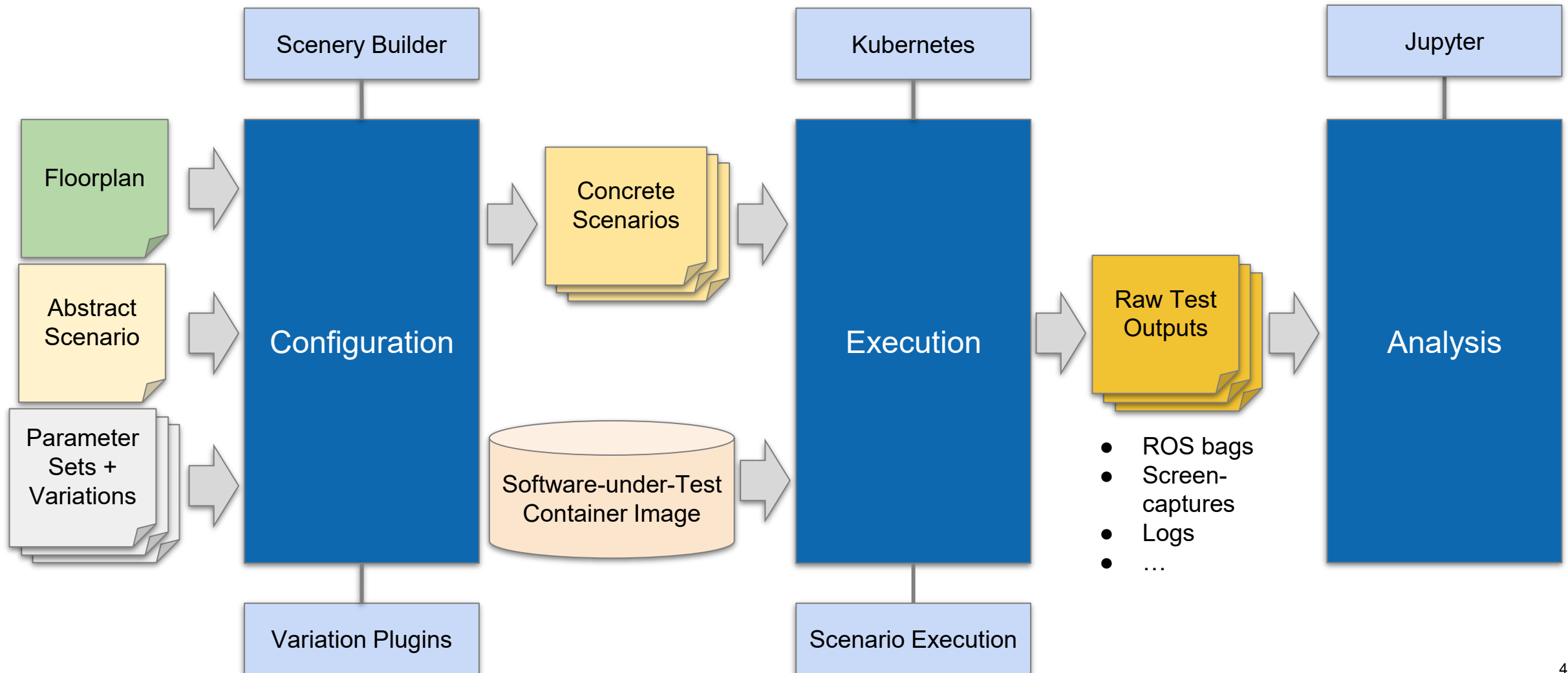
Robot Testing

- Variations: Floorplans, Paths, Obstacles, Noise, Lists and Distributions

Utility Toolkit

- Graphical Configuration Editor
- Extendible Result Viewer
- Support for multiple Kubernetes Providers

Workflow



Basic Navigation Test

```
scenario test_scenario:
  robot: differential_drive_robot
  goal_pose: pose_3d
  do parallel:
    ros_launch('nav2_bringup', 'tb4_simulation_launch.py',
      [key_value('headless', 'False')])
    capture_screen('capture.webm', frame_rate: 2.0)
  serial:
    robot.init_nav2(pose_3d())
    bag_record(['/tf', '/tf_static', '/amcl_pose',
      '/scenario_execution/snapshots'],
      use_sim_time: true,
      timestamp_suffix: false)
    robot.nav_to_pose(goal_pose)
  emit end
```

OpenSCENARIO

```
configuration:
- name: test
  scenario_file: scenario.osc
  variations:
  - ParameterVariationList:
    name: goal_pose
    values:
    - position:
      x: 21.0
      y: 6.0
execution:
  image: ghcr.io/cps-test-lab/robvast:latest
  kubernetes:
    resources:
      cpu: 6
  runs: 5
  test_files_filter:
  - "**/files/*"
analysis:
  preprocessing:
  - ../../tools/docker_exec.sh rosbags_tf_to_csv.py
  - ../../tools/docker_exec.sh rosbags_bt_to_csv.py
  visualization:
  - Analysis:
    single_test: analysis/analysis_single_test.ipynb
    config: analysis/analysis_config.ipynb
    run: analysis/analysis_run.ipynb
```

RoboVAST Configuration

Terminal

\$

Kubernetes

K9s v0.50.6 gke_aqueous-tesla-459312-c

— pods(default)[0] —

NAME	PF	READY	STATUS	RESTARTS	CPU
------	----	-------	--------	----------	-----

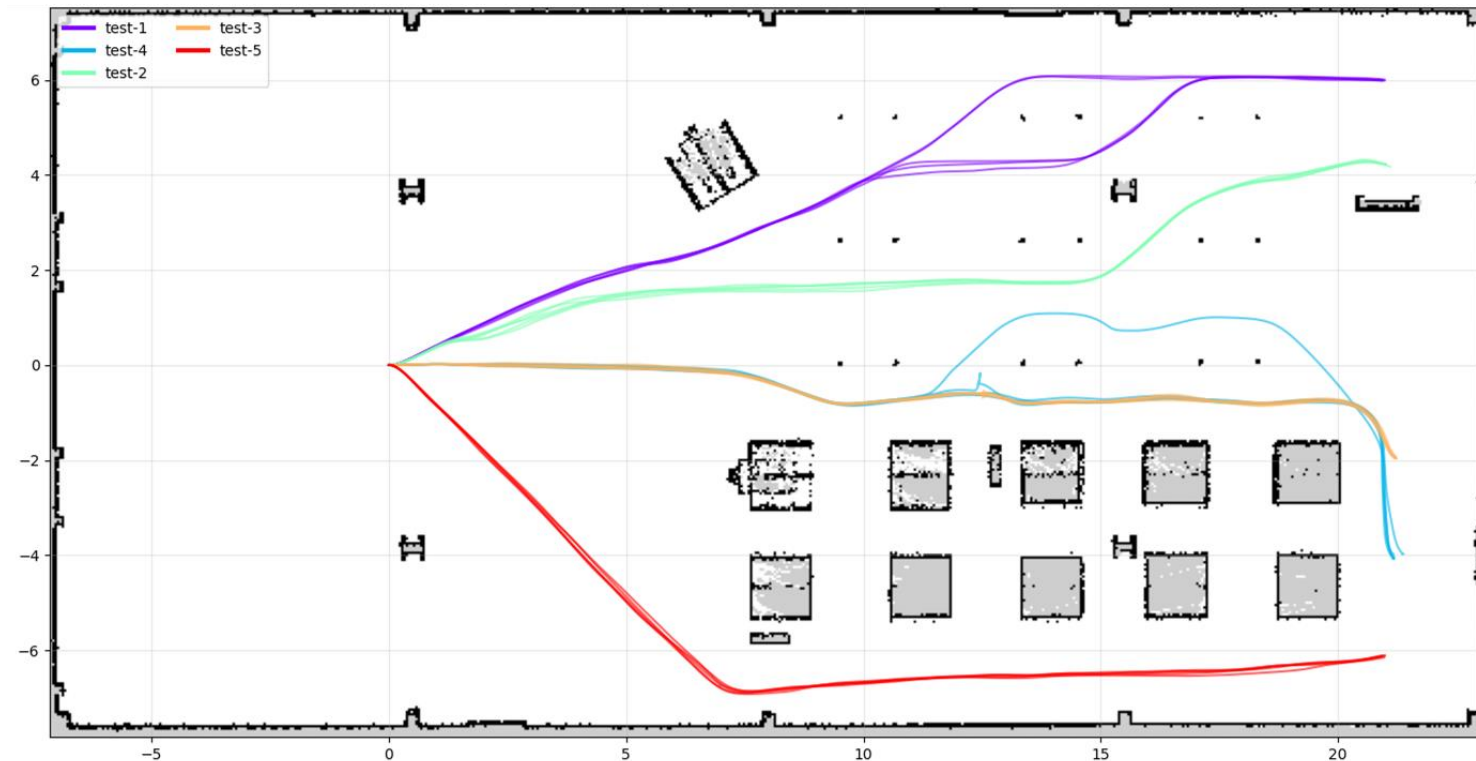
<pod>

Timer

00m 00s

Basic Navigation Test Results

 Tests in parallel	25
 Total Distance	573m
 Duration	3m 22s
 Execution on Cluster	168 vCPUs
 Costs on Hyperscaler	~0,50€



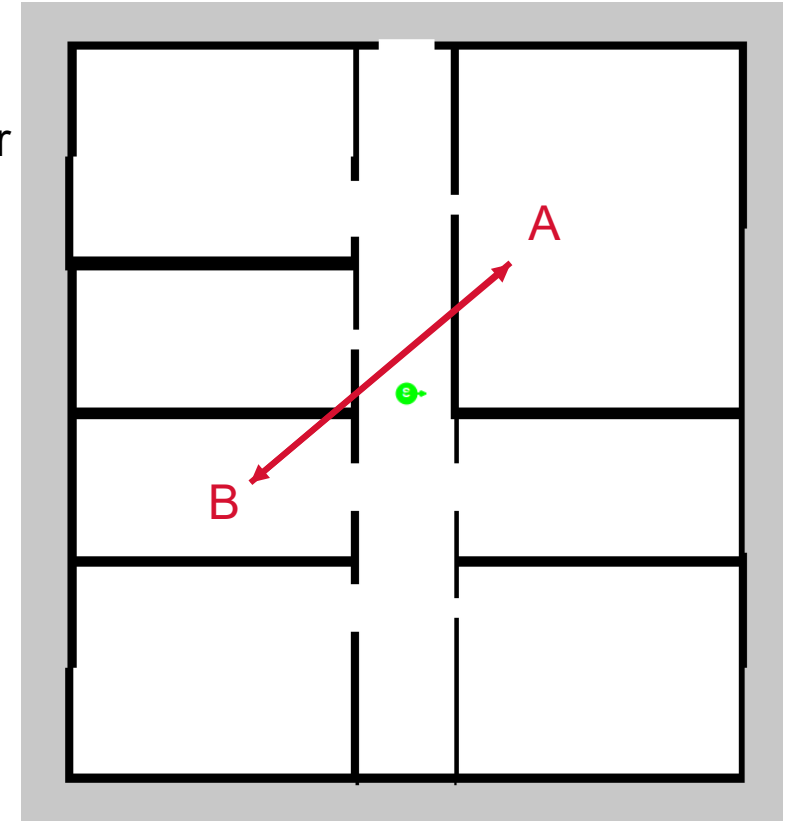
Applying RoboVAST

To check assumptions

- Are the **default NAV2 parameters** sufficient for our environment?
- Can obstacles and entryways be navigated as long as the **robot footprint can fit** between them?

To check performance

- How do obstacles affect **localization**?
- Do **metamorphic relations** hold?



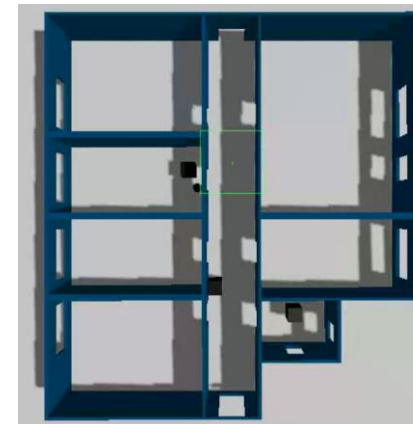
Dataset

Purpose

- Benchmark for indoor navigation using NAV2
 - Check developer [assumptions](#)
 - Check correctness of NAV2 [parameterization](#)

Structure

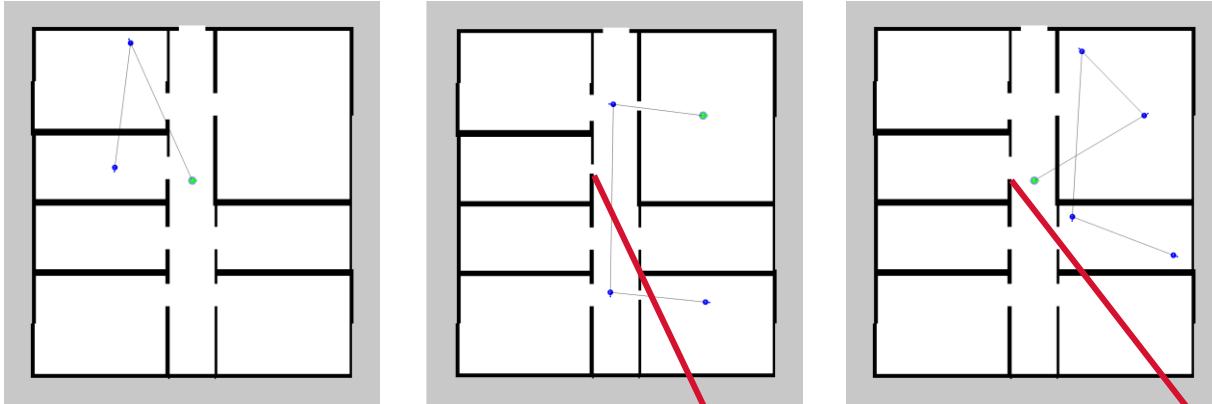
- [300](#) unique tests across various test features, [3](#) runs each, [900](#) total
- Both planned successes and planned failures, split at a [60/40](#) ratio



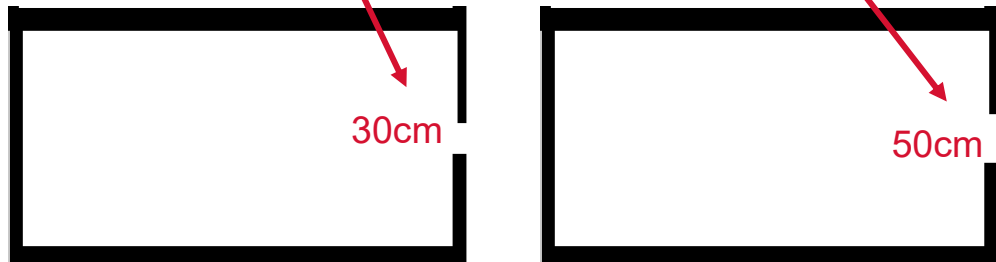
Test Results	
small-dataset-maps-door-size-0-5-1s2-1	✓ 0 ✓ 1 ✓ 2
small-dataset-maps-door-size-0-5-1s2-10	✓ 0 ✓ 1 ✓ 2
small-dataset-maps-door-size-0-5-1s2-11	✓ 0 ✓ 1 ✓ 2
small-dataset-maps-door-size-0-5-1s2-12	✓ 0 ✓ 1 ✓ 2
small-dataset-maps-door-size-0-5-1s2-13	✗ 0 ✓ 1 ✓ 2
small-dataset-maps-door-size-0-5-1s2-14	✓ 0 ✓ 1 ✓ 2
small-dataset-maps-door-size-0-5-1s2-2	✓ 0 ✗ 1 ✗ 2
small-dataset-maps-door-size-0-5-1s2-3	✗ 0 ✗ 1 ✗ 2
small-dataset-maps-door-size-0-5-1s2-4	✗ 0 ✗ 1 ✗ 2
small-dataset-maps-door-size-0-5-1s2-5	✓ 0 ✗ 1 ✓ 2
small-dataset-maps-door-size-0-5-1s2-6	✗ 0 ✗ 1 ✗ 2
small-dataset-maps-door-size-0-5-1s2-7	✗ 0 ✗ 1 ✗ 2
small-dataset-maps-door-size-0-5-1s2-8	✗ 0 ✓ 1 ✓ 2
small-dataset-maps-door-size-0-5-1s2-9	✓ 0 ✓ 1 ✓ 2

Variations

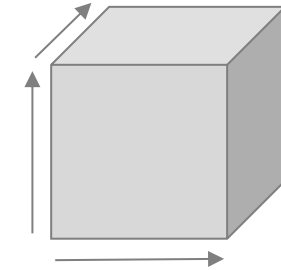
Waypoints



Door widths

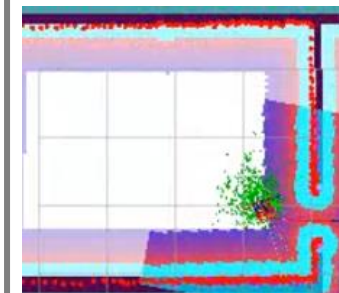


Obstacle Size

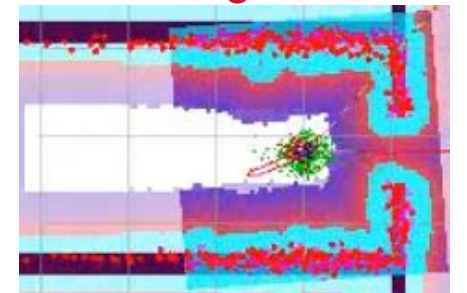


LiDAR Noise

Low

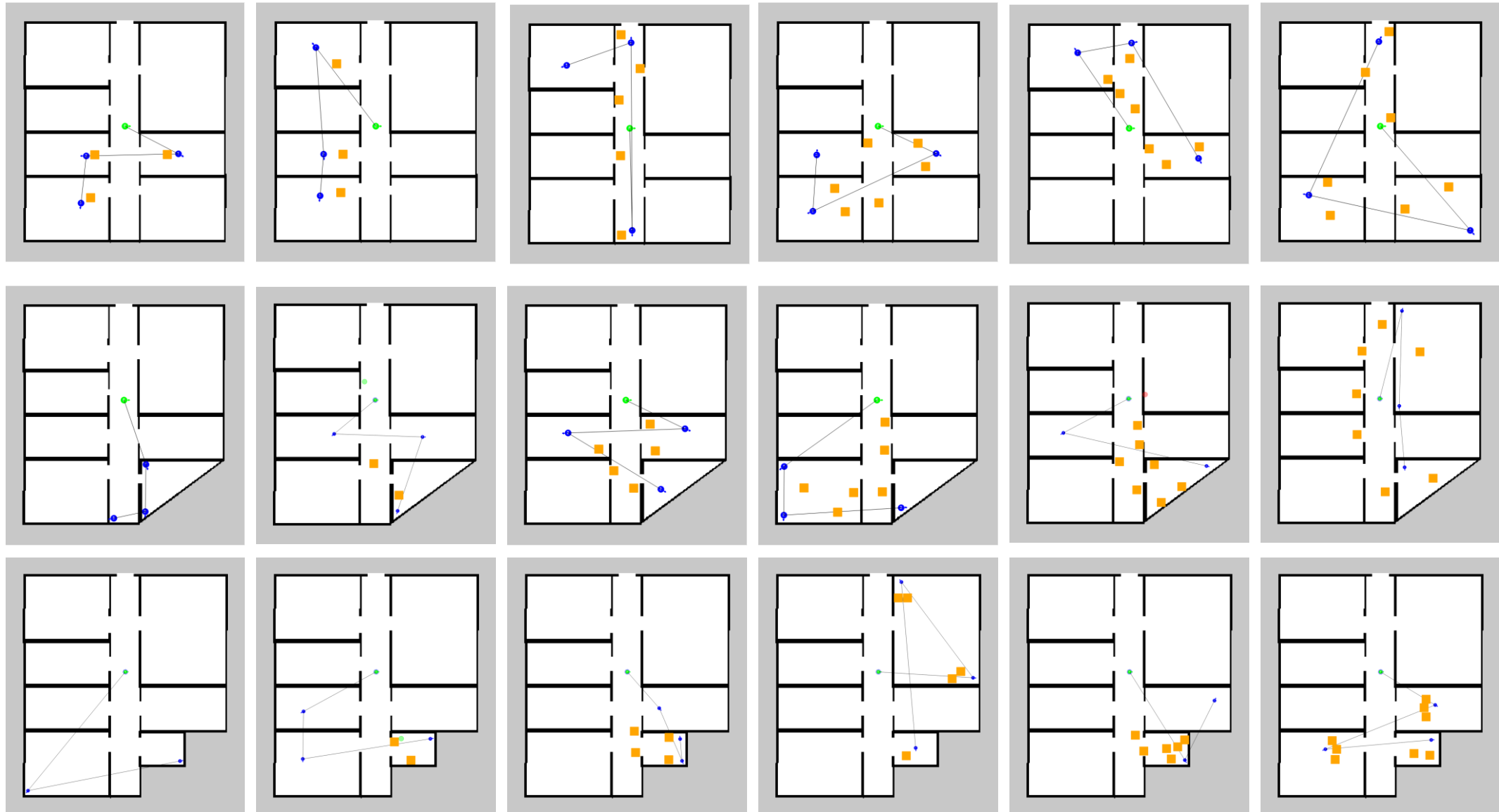


High



Obstacle Count

Room Layout



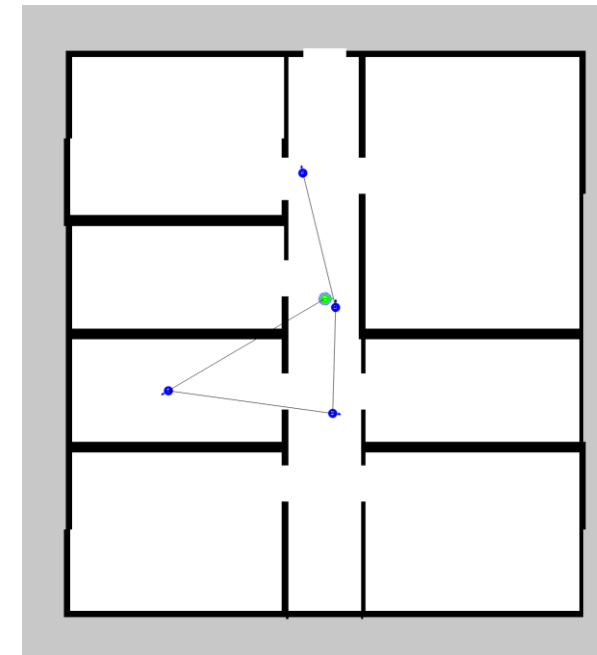
Constructing the Dataset

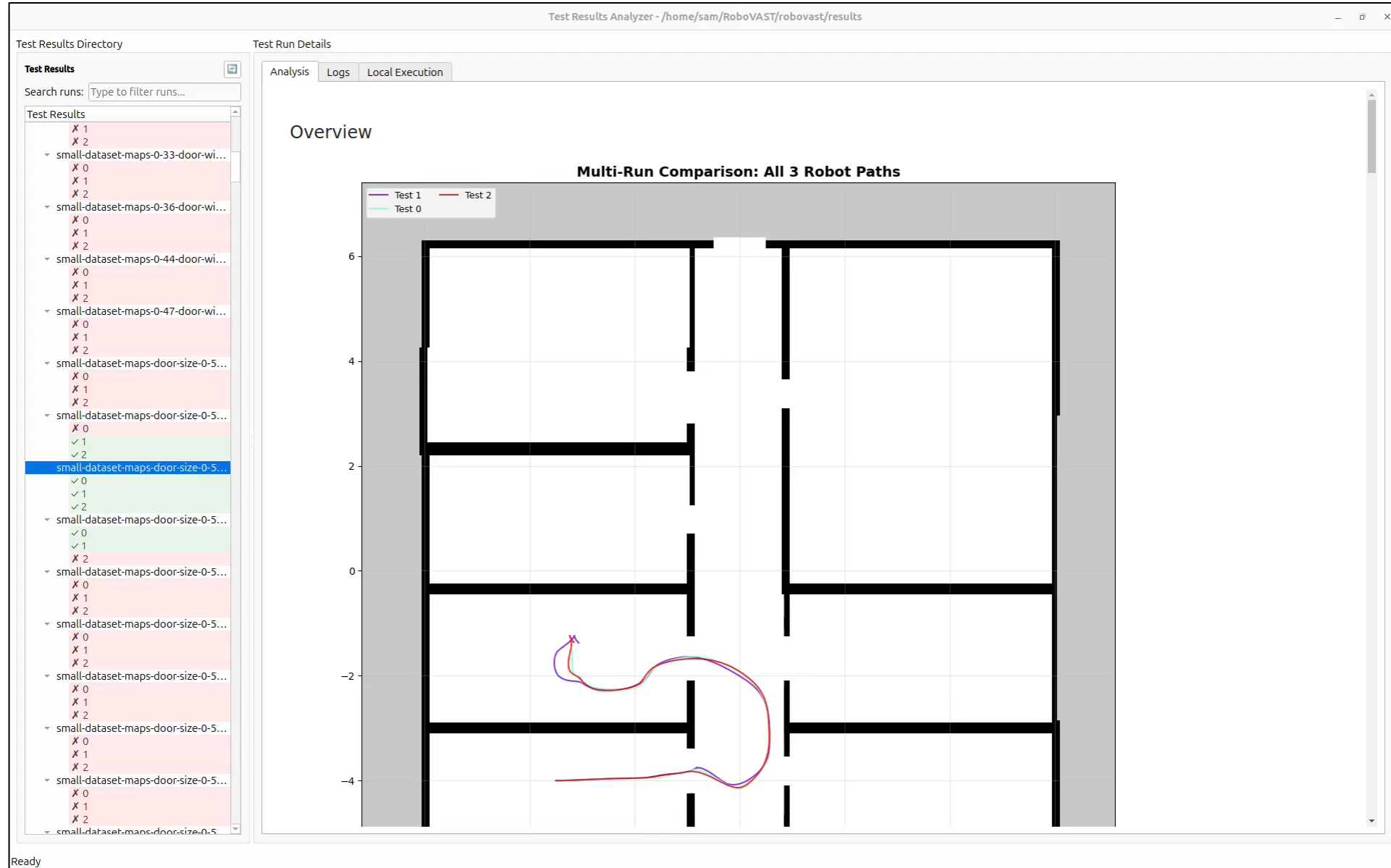
Determine variation dimensions

- Variations chosen based on **known failure causes**
- Waypoints and obstacles both randomly and hand-placed to **induce failure**
- Noise and map layout assigned randomly

Expected failure causes

- Obstacles shorter than LiDAR height
- Doors too narrow
- Waypoints outside reachable area
- LiDAR noise prevents obstacle differentiation
- Obstacles blocking path
- Goal waypoint near intermediate waypoint

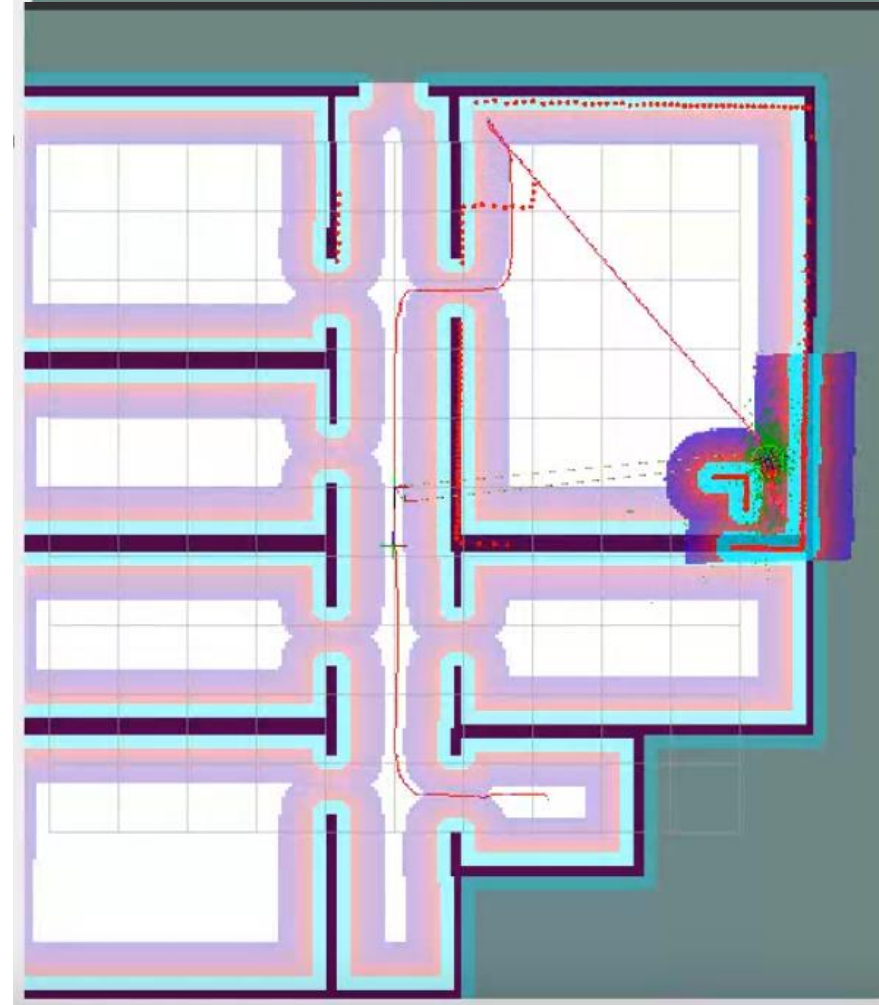




Analyzing the Dataset

Unplanned failures

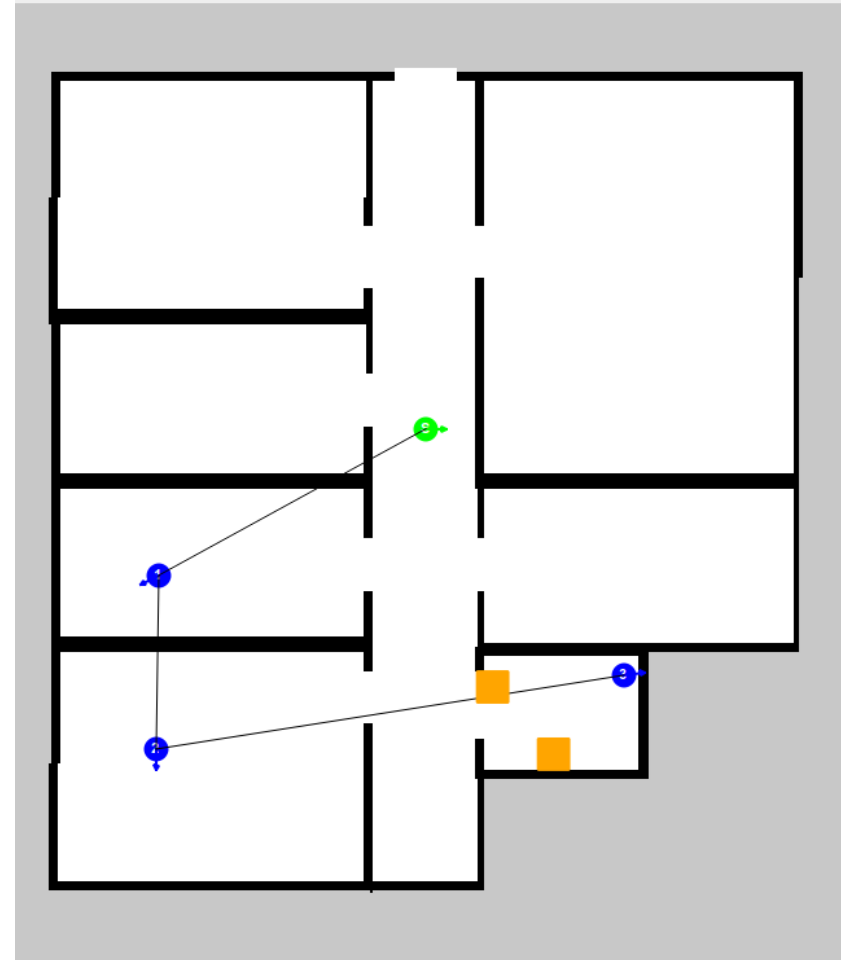
- Unreachable waypoints due to high inflation radius
- Robot became stuck *returning* past obstacles
- Robot was caught on an obstacle



Analyzing the Dataset

Unplanned successes

- Waypoints that were intended to be unreachable were creatively solved
- Obstacles that were intended to block waypoints didn't consider goal threshold
- Goal reached despite high sensor noise and obstacles



Conclusions from the Dataset

- Ensure that Nav2 parametrization considers the **assumptions** behind the robot footprint and the environment
- Do not expect **metamorphic relations** to hold
- Ensure **critical constraints** are adhered to
- Random failures can also mean **random successes**

Questions?



Test
Variation in
Robotics is
too much effort.

With
RoboVAST
I just do it!



<https://github.com/cps-test-lab/robovast>