Breaking Bots: Robustness Testing for ROS

ROSCon ’23, New Orleans, USA

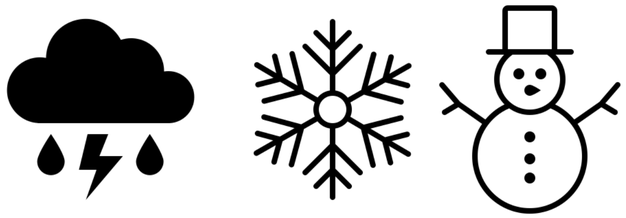
Chris Timperley, October 19th, 2023

Carnegie Mellon University

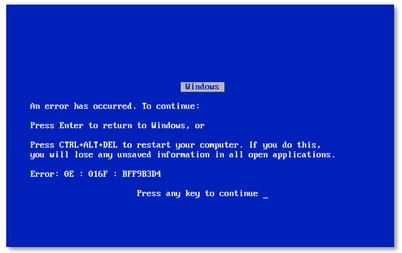
Robustness Testing

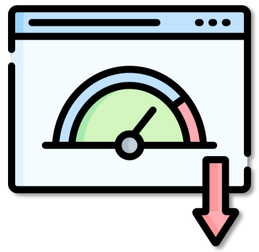
Ability to test systems under rare, off-nominal, extreme conditions

Environmental

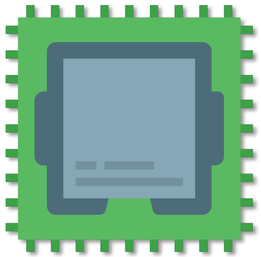


Software





Hardware

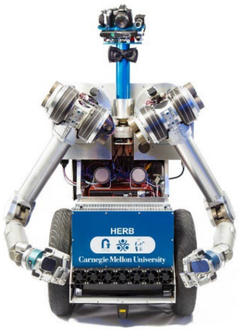


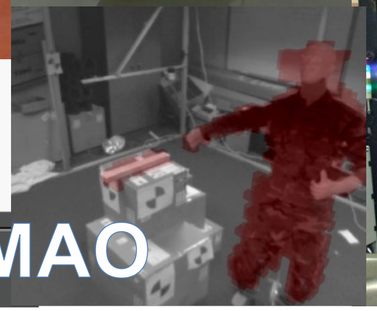
When things break down and assumptions are violated, the system must continue to be safe!

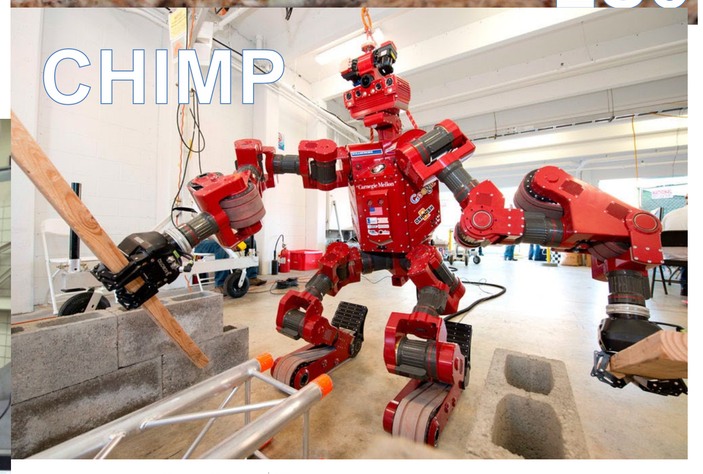
Robustness Testing at NREC

NREC
EXPERIENCE · EXPERTISE · EXCELLENCE

Tested dozens of government and commercial systems since 2011







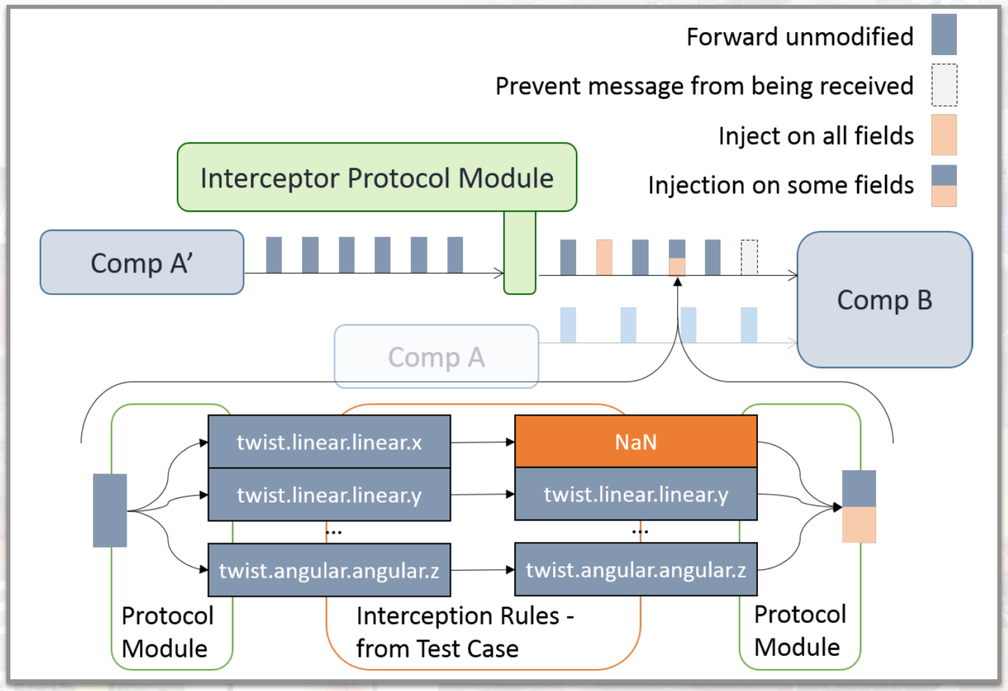
ROS

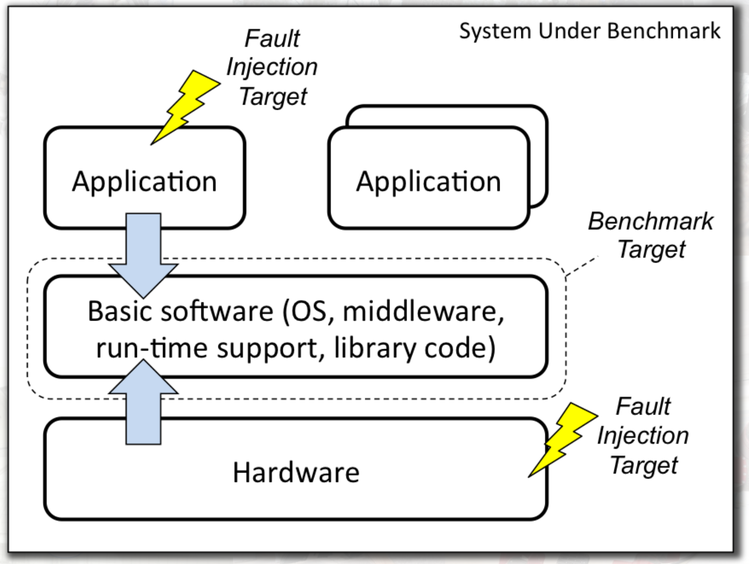
Inter-Process Communication

SAE J1939

Robustness Testing at NREC

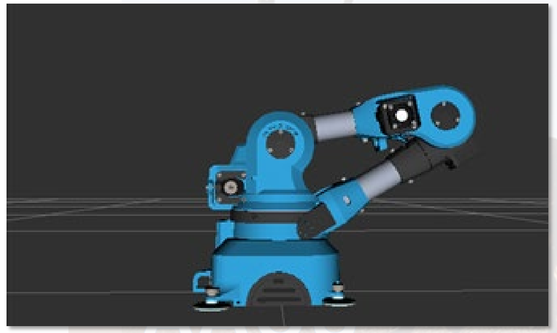
Built a suite of robustness tools that target different interfaces





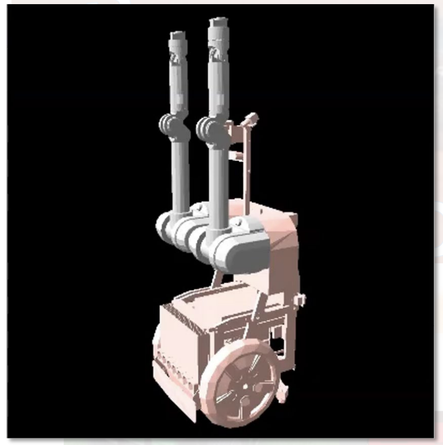
Robustness Testing at NREC

Found and helped to fix bugs in over 30 systems that were tested



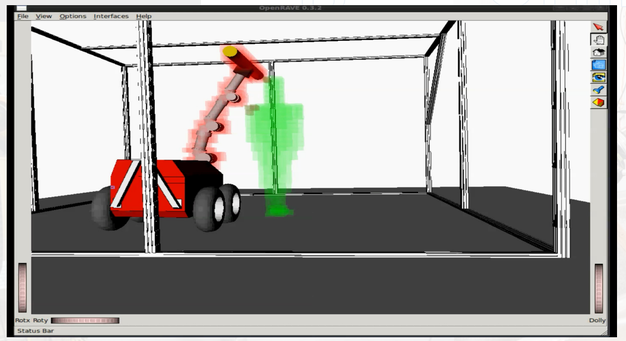
In Simulation
Message on /joy topic causes self-intersection

In Hardware
Spurious speed command causes speed limit violation



In The Wild
test value caused self-intersection

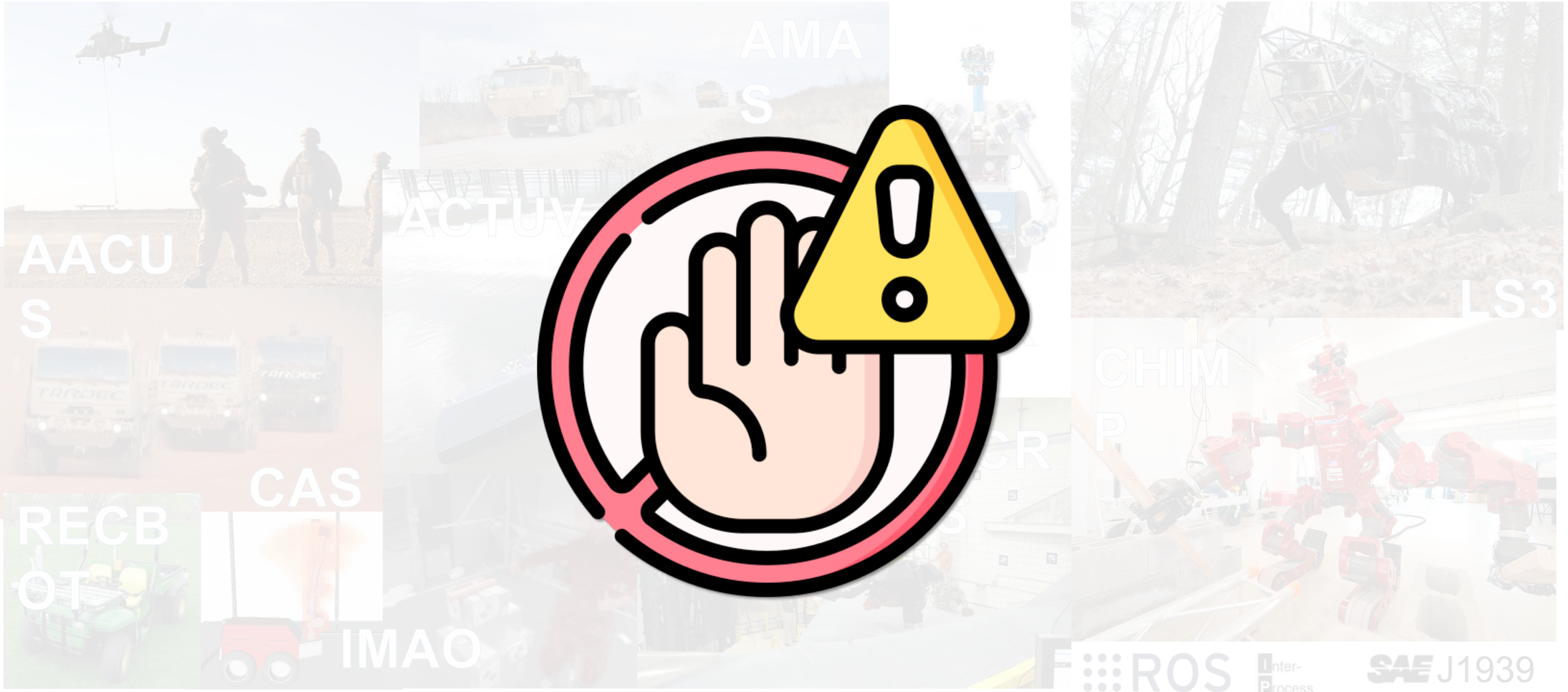
months later, failure occurred on physical robot and caused irreparable damage



With Human Harm
Spurious joint angle causes invalid state and stops safety system from functioning

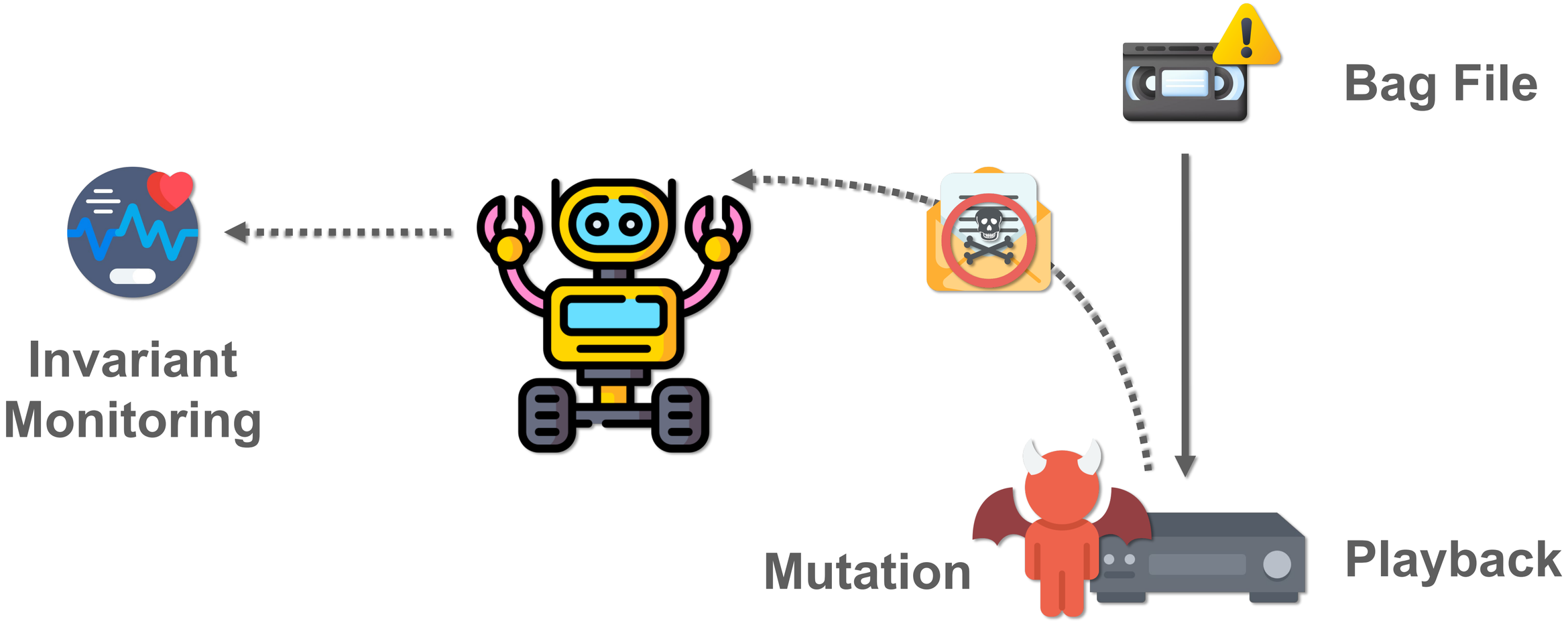
Robustness Testing at NREC

But almost all of this work was done under restricted funding



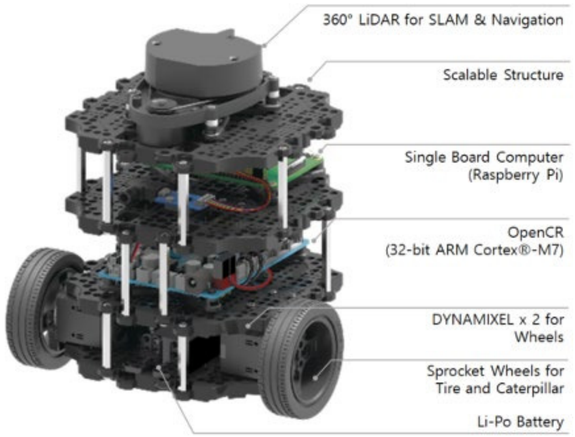
MOBSTA: Robustness Testing for ROS

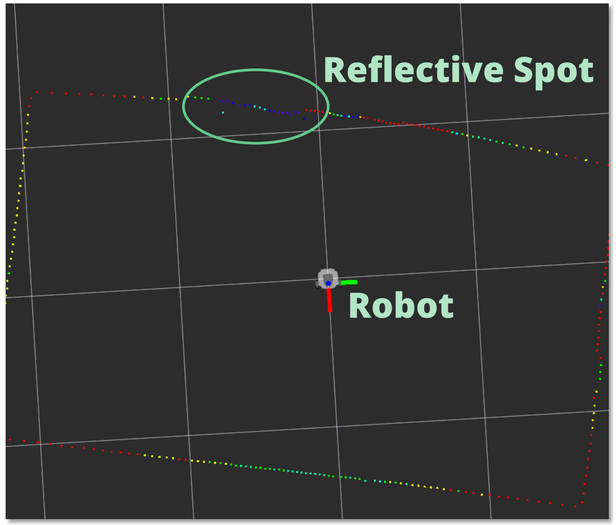
Early Research Release

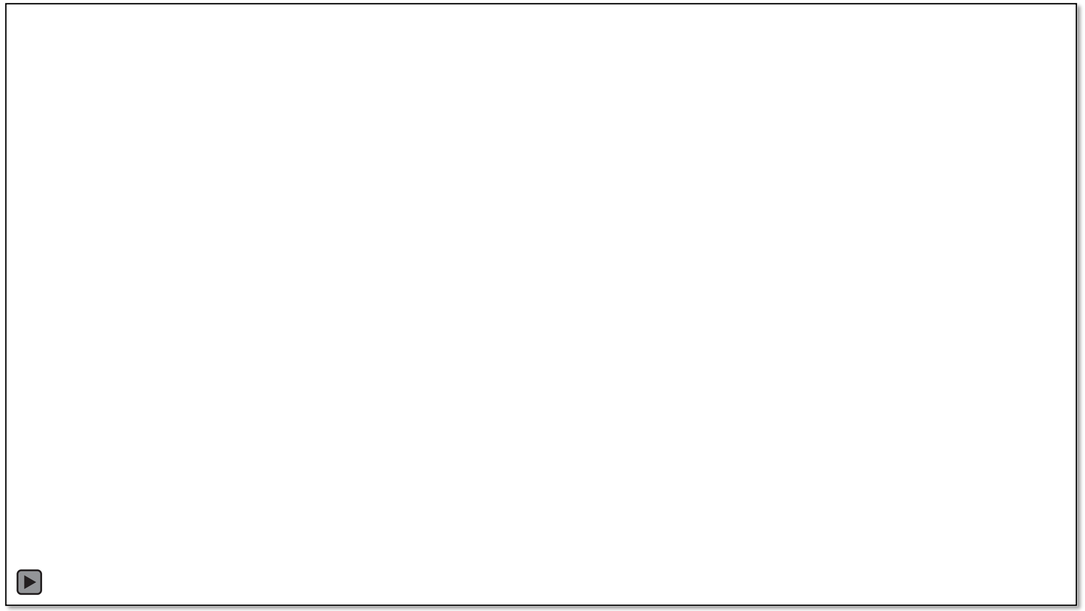


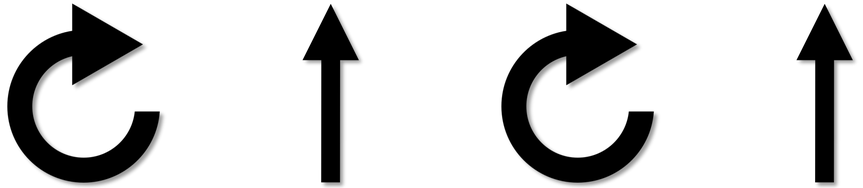
Example System: TurtleBot

Automatic Lidar Parking









MoBSTA: Robustness Testing for ROS

Three Ingredients for Robustness Testing

Nominal Data

Invariants

Mutations

TurtleBot Automatic Lidar Parking

Nominal Field Data Collection

- Setup an enclosed rectangular environment for the robot
- We vary the position of the the reflective tape, used to indicate the parking spot
- We vary the starting position of the robot
- Collected a total of 10 log files

TurtleBot Automatic Lidar Parking

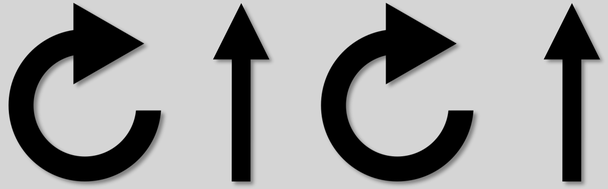
Invariants

Valid Cmd Vel

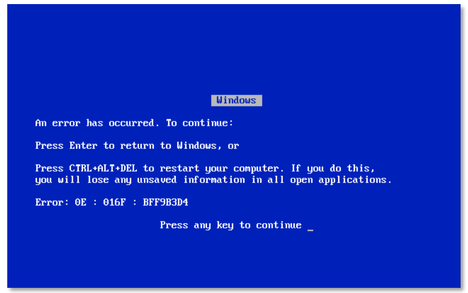
```
invariant_name: ValidNumbersInvariant
invariant_params:
  velocity_topic: /cmd_vel
```

Command Timing

```
invariant_name: CorrectCommandsInvariant
invariant_params:
  velocity_topic: /cmd_vel
  joint_states_topic: /joint_states
  bad_command_limit: 10
  angle_goal_deg: 5
  position_goal: 0.05
  distance_goal: 0.5
  angle_uncertainty_deg: 3
  position_uncertainty: 0.2
  parking_spot_x_1: -1.25
  parking_spot_y_1: -0.623
  parking_spot_x_2: -0.825
  parking_spot_y_2: -0.754
```



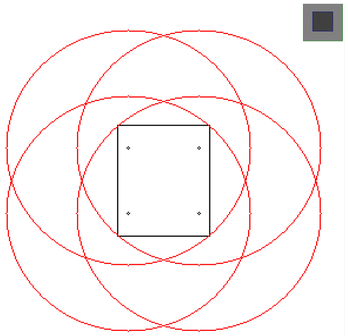
MoBSTA supports a wider set of invariants



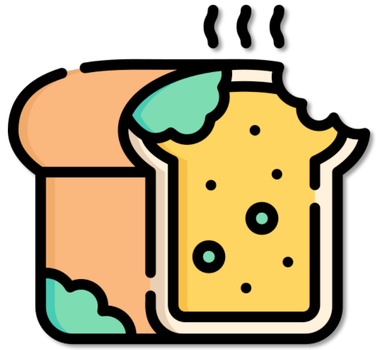
Node Crashes

Timeliness

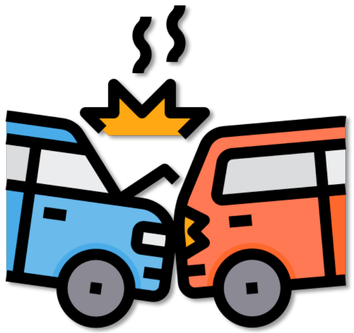
Speed Limits



Turning Radius



Staleness



Unsafe Plans

TurtleBot Automatic Lidar Parking

Mutation: Lidar Intensities

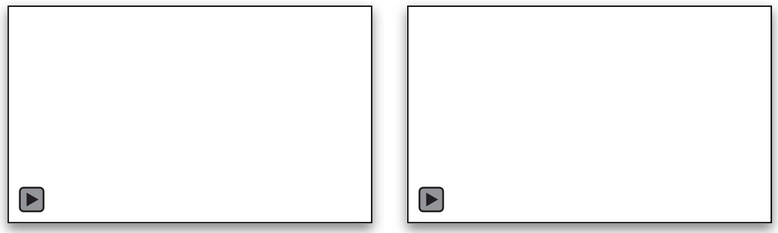
```
mutation-config:
 topics:
  /scan:
    - message_intercept: /intensities
      mutations:
       - mutation_type: Float32Array_AddToWholeArrayMutator
         mutation_chance: 1
         timeframe_begin: 0
         timeframe_end: -1
         mutation_args:
          valueToAdd: -100
          minArrayValue: 0
          maxArrayValue: 255
          arraySize: 259
```

sensor_msgs/LaserScan

std_msgs/Header header
float32 angle_min
float32 angle_max
float32 angle_increment
float32 time_increment
float32 scan_time
float32 range_min
float32 range_max
float32[] ranges
float32[] intensities

We support other realistic perturbations

Sensor inputs, control and planning data, configuration



Dust on Lens

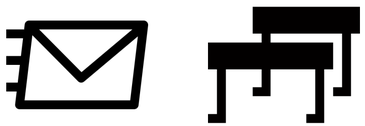
Compression

Camera Distortion

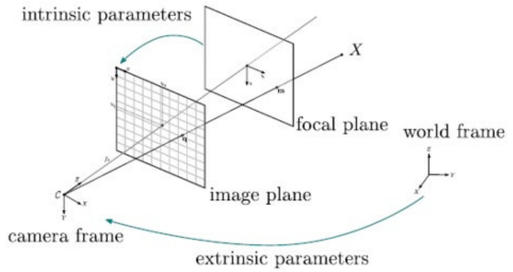
Exposure

NavSatStatus status	STATUS_SBAS_FIX | SERVICE_GPS
Float64 latitude	**40°28'31.8" N**
Float64 longitude	**NaN° W**

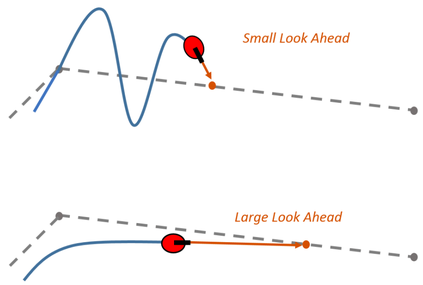
Field Mutation



Message Loss and Delay



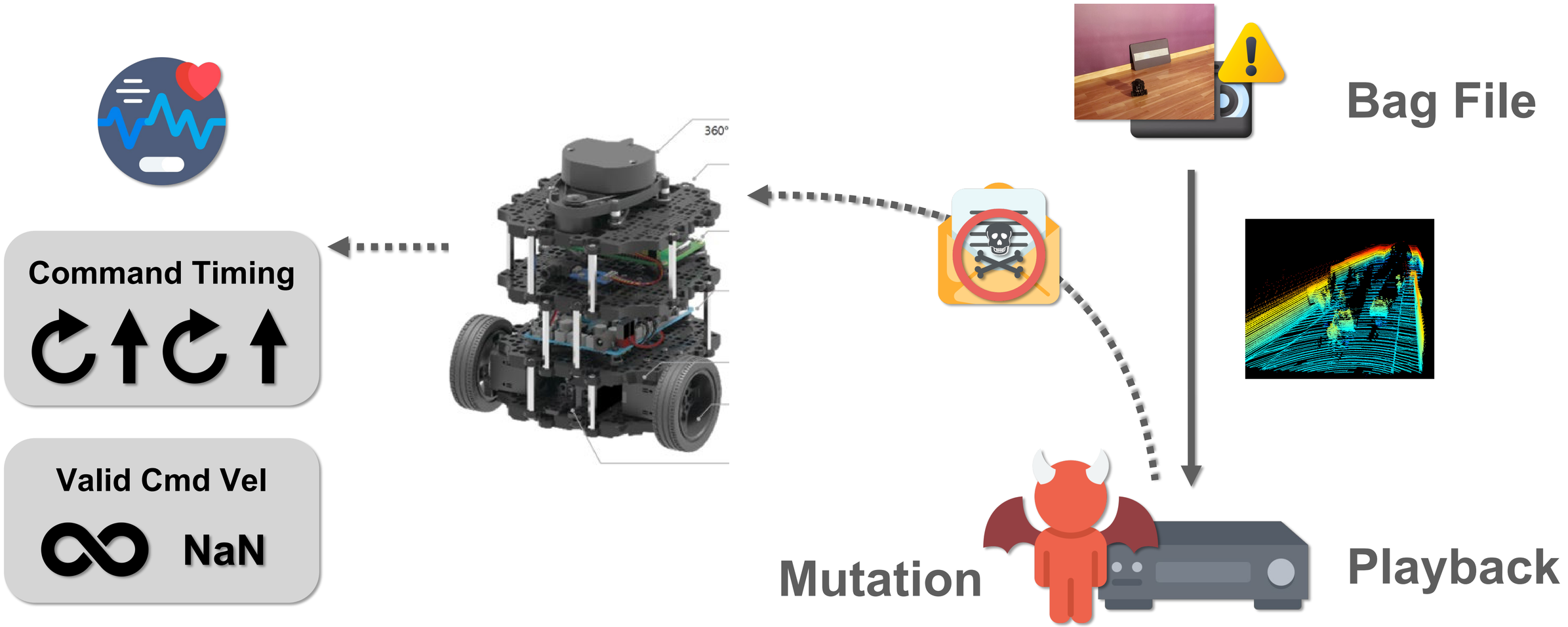
Camera Parameters



Algorithm Parameters

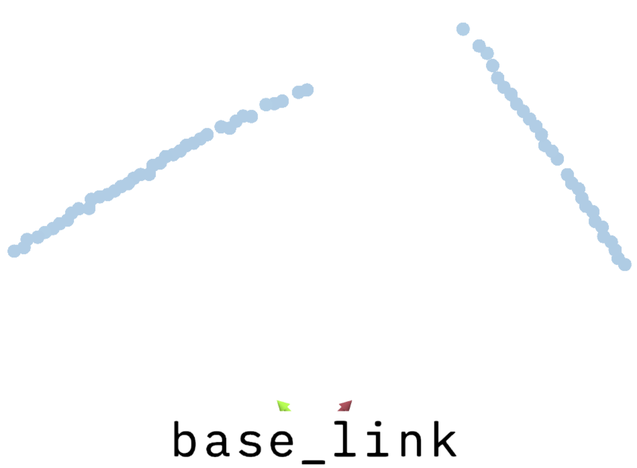
TurtleBot Automatic Lidar Parking

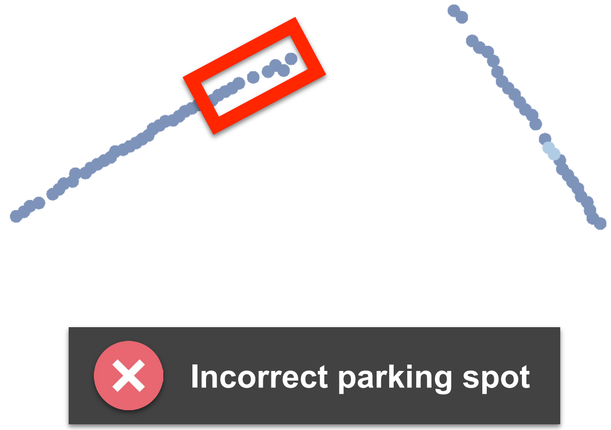
End-to-End Overview

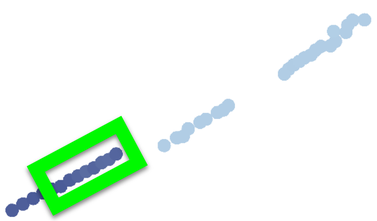


TurtleBot Automatic Lidar Parking

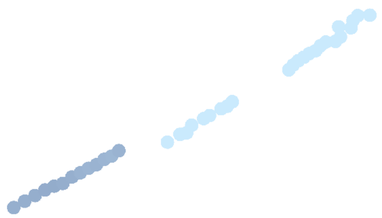
Mutation exposes brittle lidar intensity handling



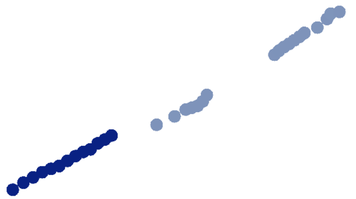




Nominal



-100



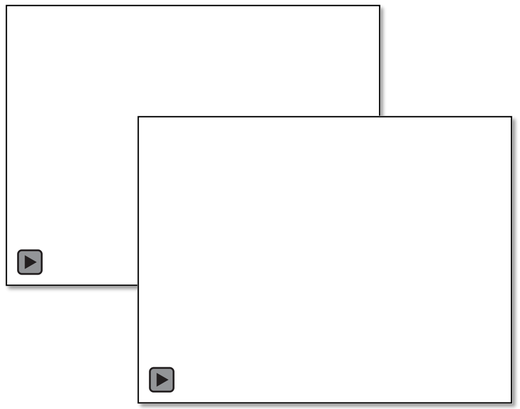
+100

TurtleBot Automatic Lidar Parking

Mutation exposes brittle lidar intensity handling

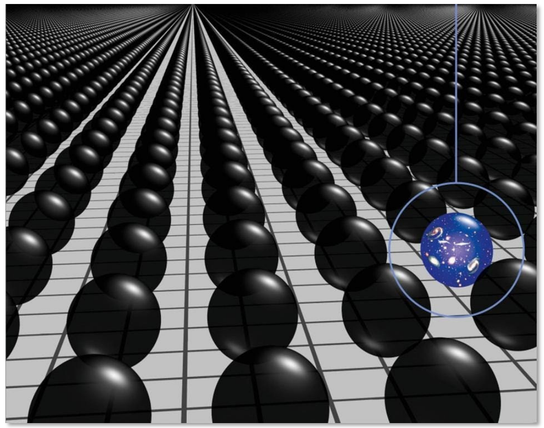
```
intensity_threshold = 100
...

for i in range(len(msg.intensities)):
    spot_intensity = msg.intensities[i] ** 2 * msg.ranges[i] / 100
    ...

    if spot_intensity >= intensity_threshold:
        ...
```

What’s Next?



Closed-Loop Robustness Testing via Enhanced Simulation

AutoMOBSTA

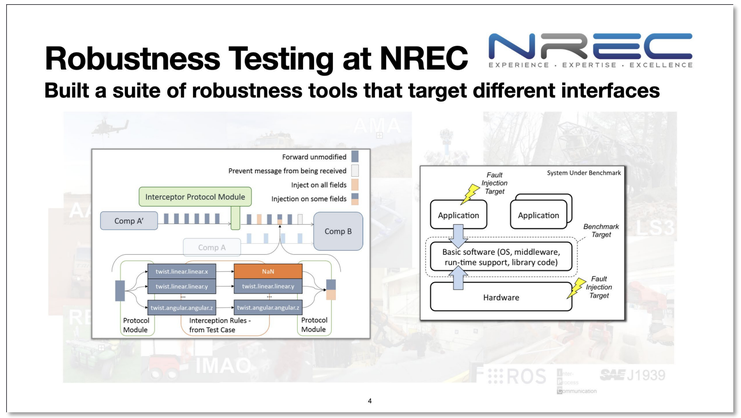


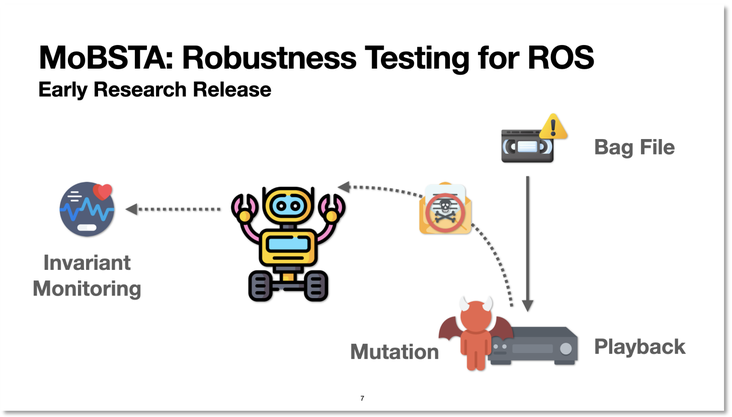
Efficient Search

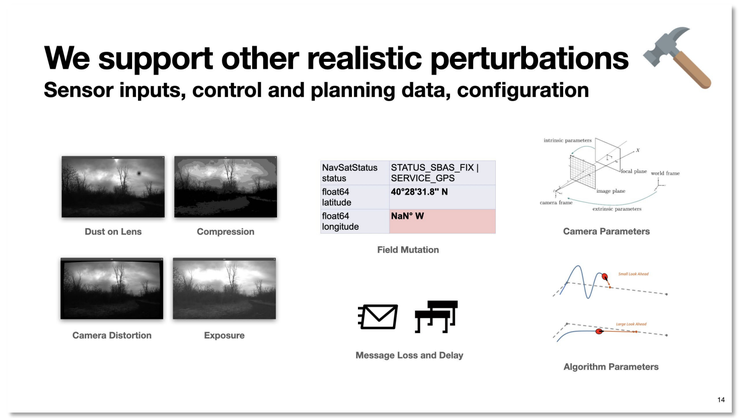
Mature Tooling + Workflow Integration

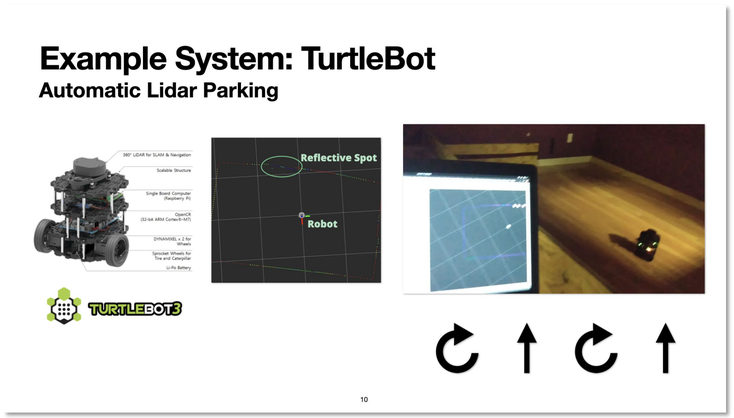
Breaking Bots: Robustness Testing for ROS

We want to hear your thoughts! Do you do robustness testing?









GitHub

SCAN ME

MoBSTA