Hardware Controller Node

Ali Mahdavifar

Background

- Autonomous Cleaning Robot

Functionalities:

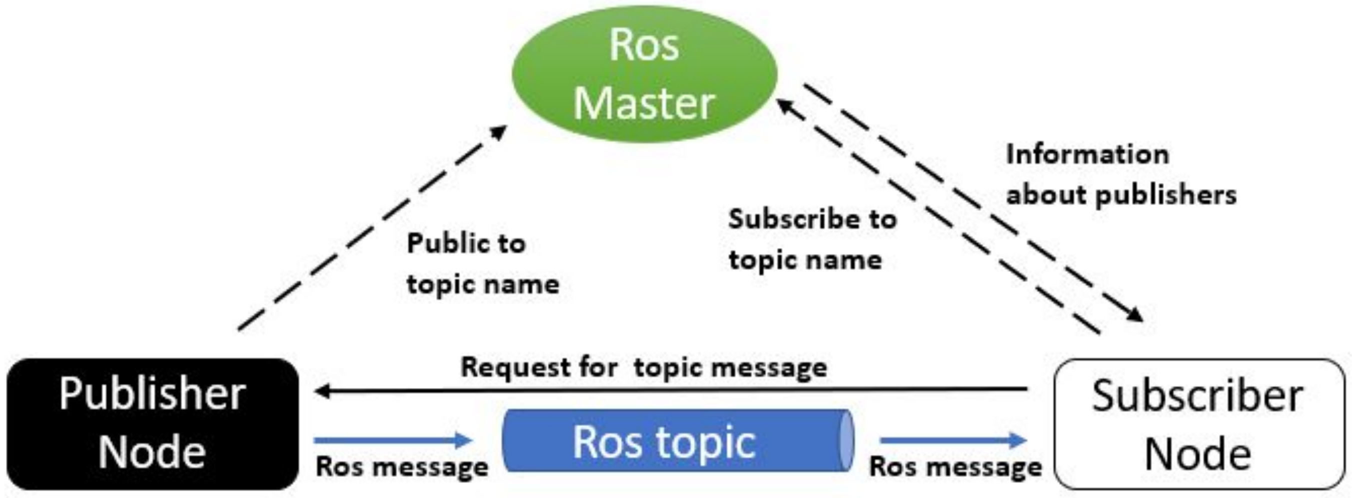
- Cleaning various surface types
- Disinfecting
- Vacuuming

Locations:

- Warehouses
- Airports
- Shopping malls
- Office areas

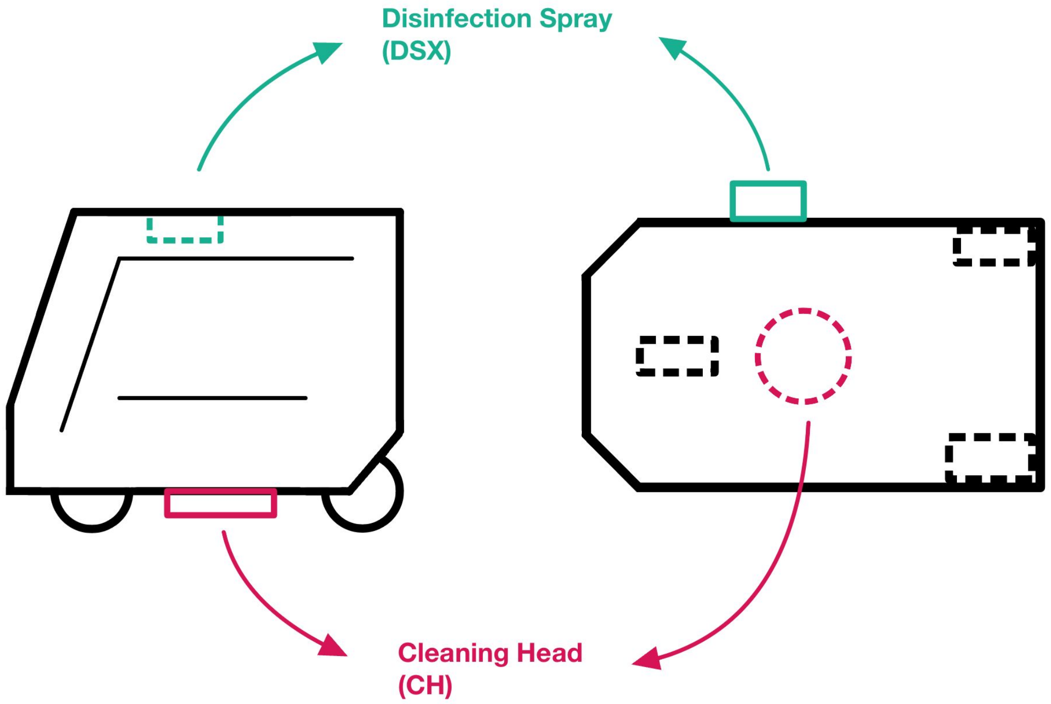
About ROS

- Robot Operating System
- An open-source framework for robot software development
- Modular software architecture
- Inter-process communications



Task

- Create a new ROS node to control robot’s cleaning hardware elements

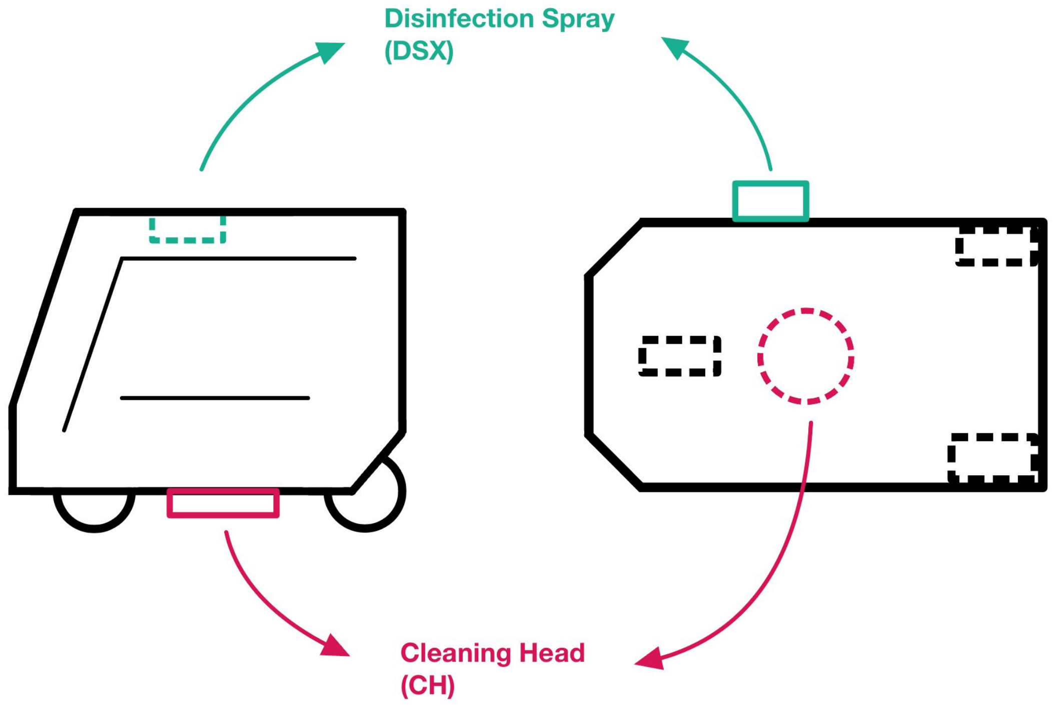


Task

- Create a new ROS node to control robot's cleaning hardware elements

DSX Mode:

- States:
  - ON: 1
  - OFF: 0
- V_cmd = dsx_speed



Task

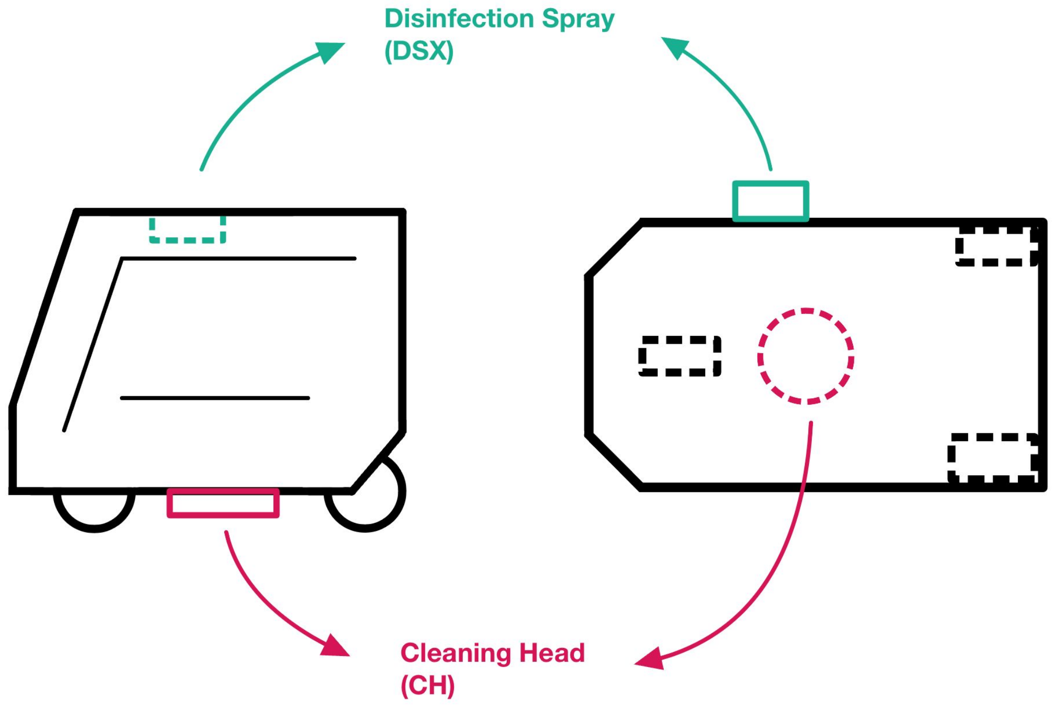
- Create a new ROS node to control robot’s cleaning hardware elements

DSX Mode:

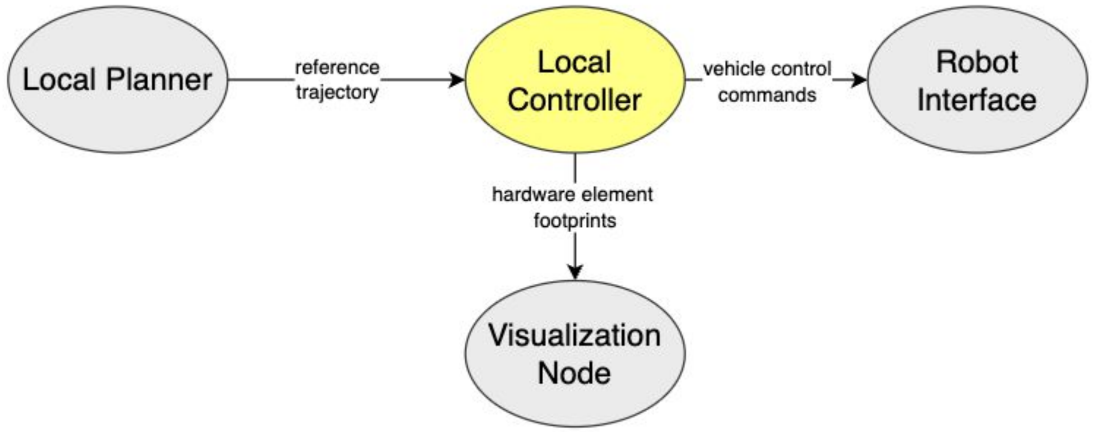
- States:
  - ON: 1
  - OFF: 0
- V_cmd = dsx_speed

Scrub Mode for CH:

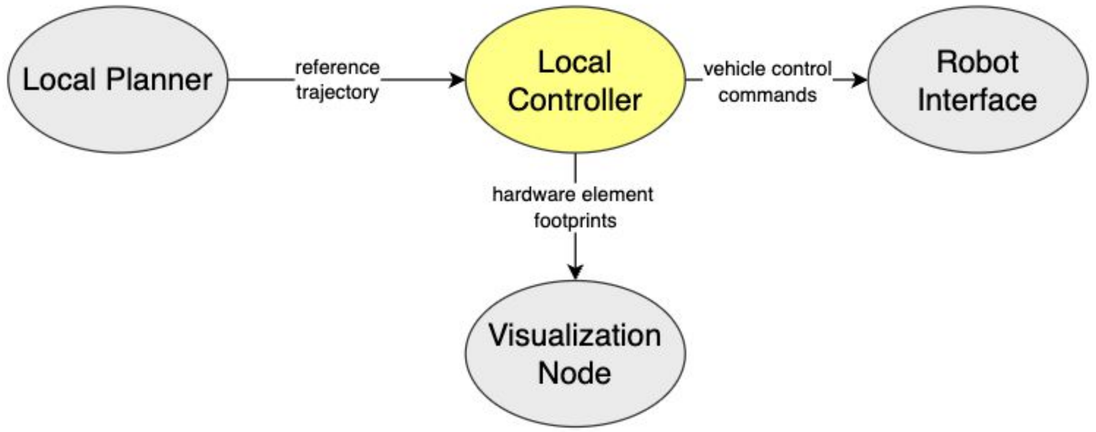
- States:
  - Lowered: 1
  - Lowering: -1
  - Raised: 0
  - Raising : -2
- V_cmd = slow_speed

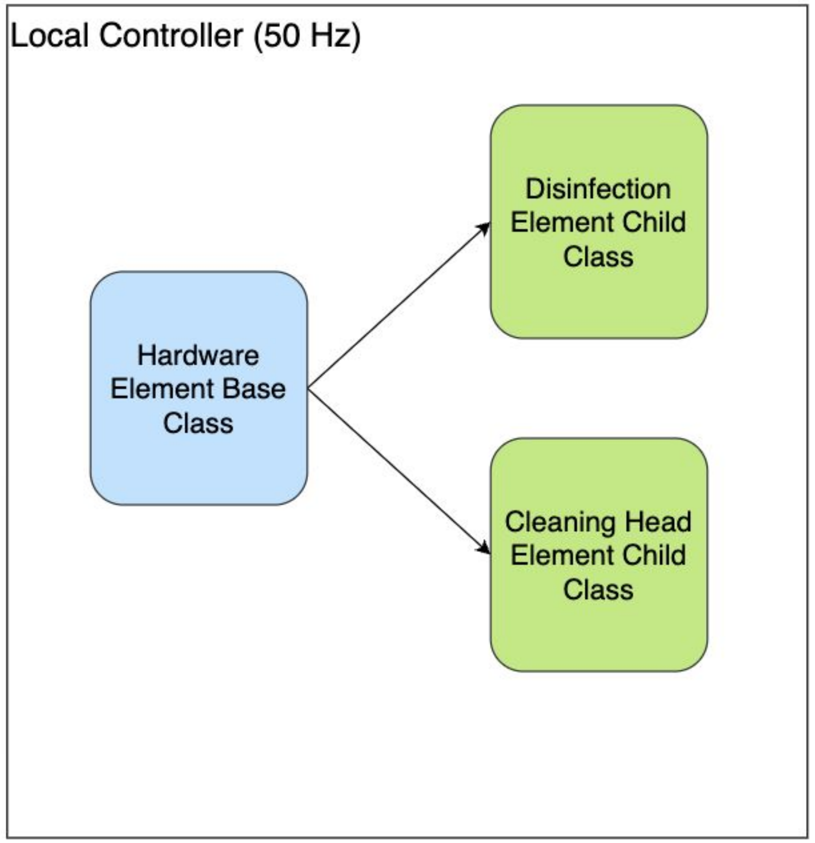


Previous Implementation



Previous Implementation




Local Controller (50 Hz)
Hardware Element Base Class
Disinfection Element Child Class
Cleaning Head Element Child Class

Previous Implementation

- Obsolete dependencies on local controller parameters

Previous Implementation

- Obsolete dependencies on local controller parameters
- LC runs at a higher frequency than LP (30-50 Hz vs. 10-20 HZ)

Previous Implementation

- Obsolete dependencies on local controller parameters
- LC runs at a higher frequency than LP (30-50 Hz vs. 10-20 HZ)
- Code extensibility issues

Requirements

- Ensuring the main motion planning functionalities are intact

Requirements

- Ensuring the main motion planning functionalities are intact
- Demonstrating reduced computational effort
    - Minimum of 5% reduction in total cleaning time

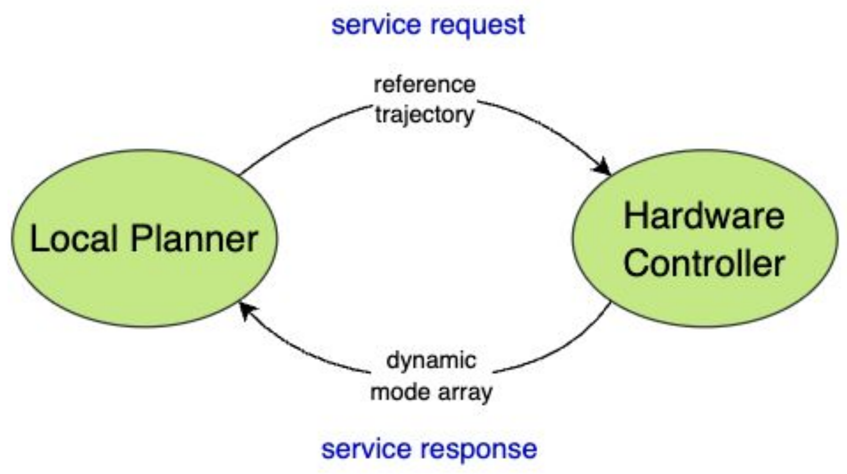
Requirements

- Ensuring the main motion planning functionalities are intact
- Demonstrating reduced computational effort
  - Minimum of 5% reduction in total cleaning time
- Addressing edge cases

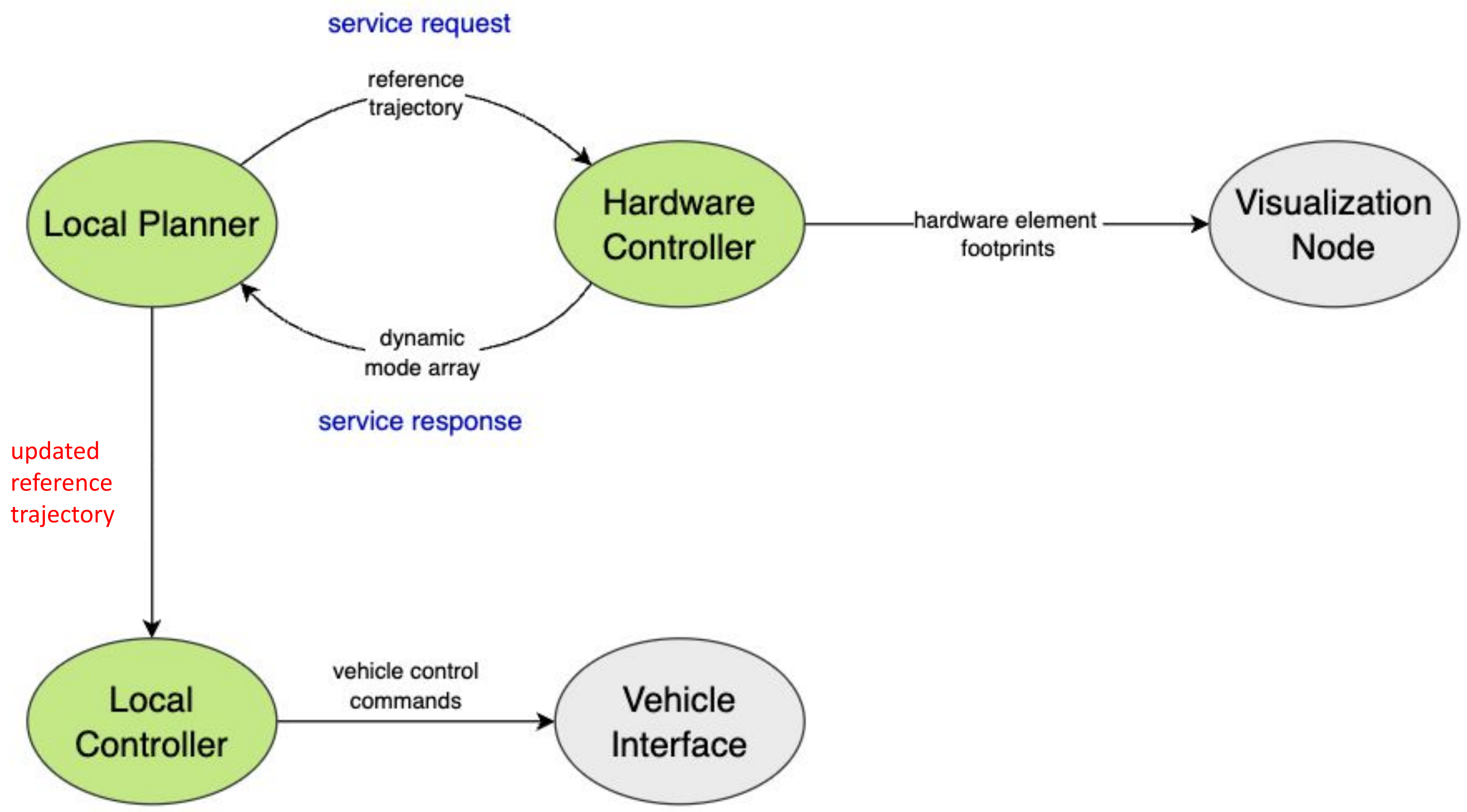
Constraints

Computational Resources	Operate within available RAM and CPU limits
Safety	Ensure a safe operation around users and passersby
Scalability	To be functional on all robot hardware versions

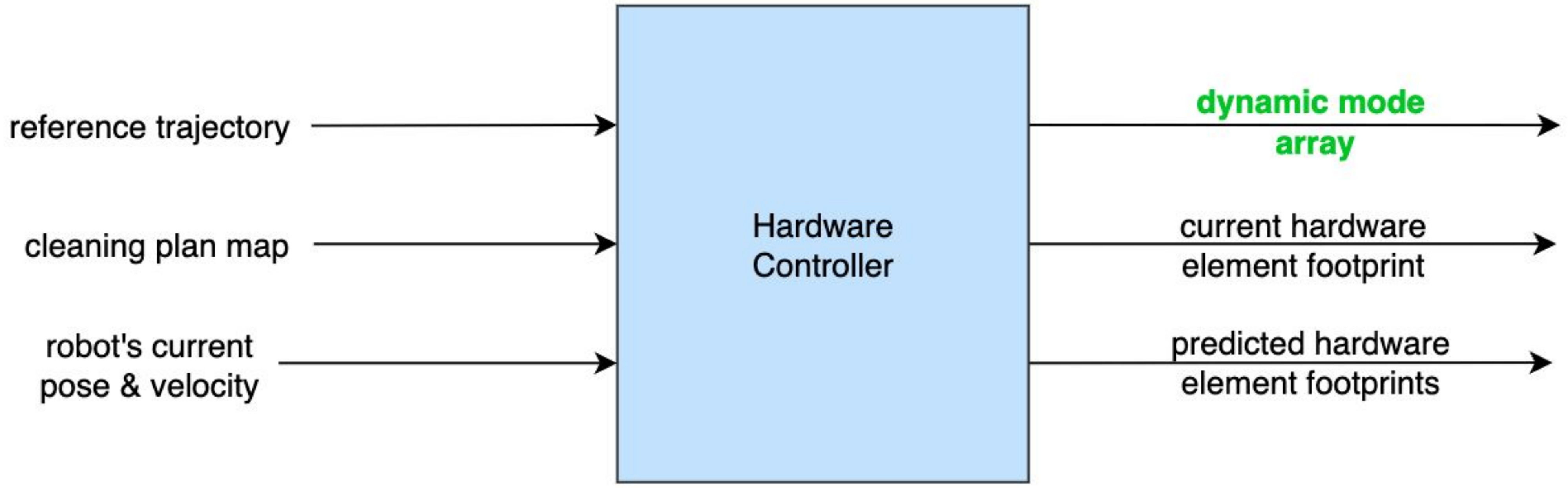
My Implementation: ROS Service-Client



My Implementation: ROS Service-Client

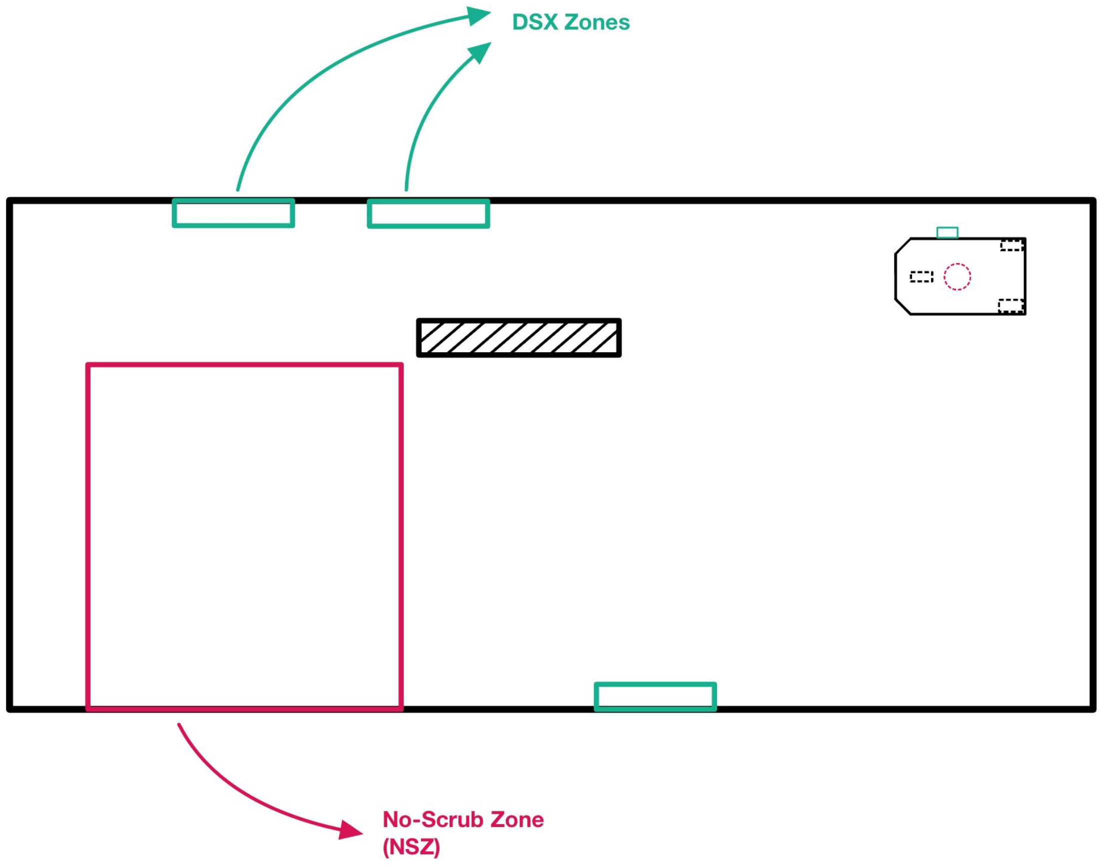


A Closer Look at HC Node



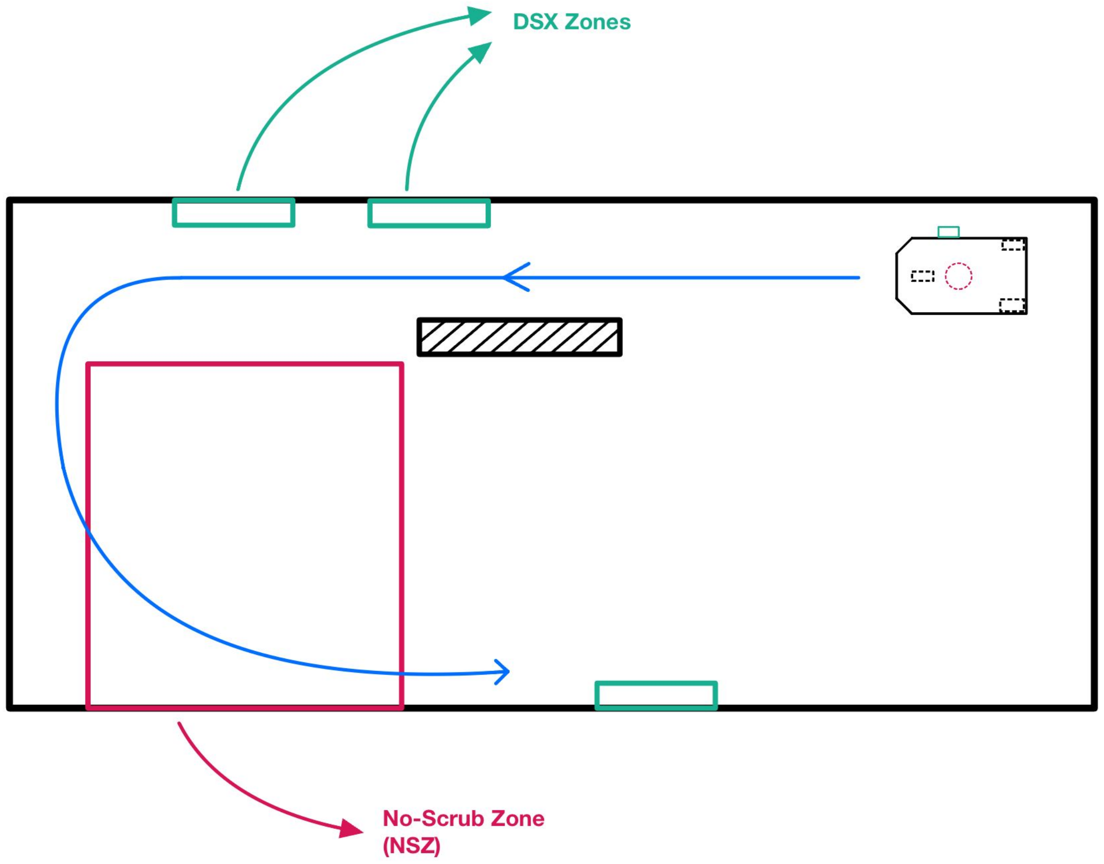
reference trajectory
cleaning plan map
robot's current pose & velocity
Hardware Controller
dynamic mode array
current hardware element footprint
predicted hardware element footprints

Cleaning Plan Layout



Map not to scale!

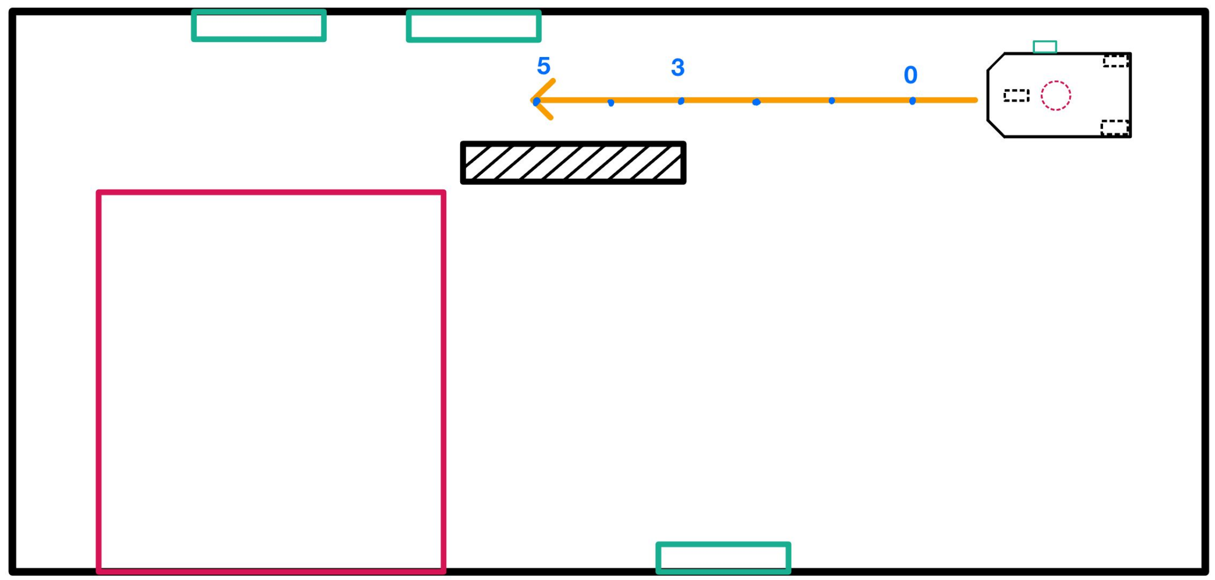
Cleaning Plan Layout



Dynamic Mode Array

- Mode: normal speed
- Mode End Index: 3
- DSX Spray State: 0
- CH State: 1

- Mode: slow down
- Mode End Index: 5
- DSX Spray State: 0
- CH State: 1

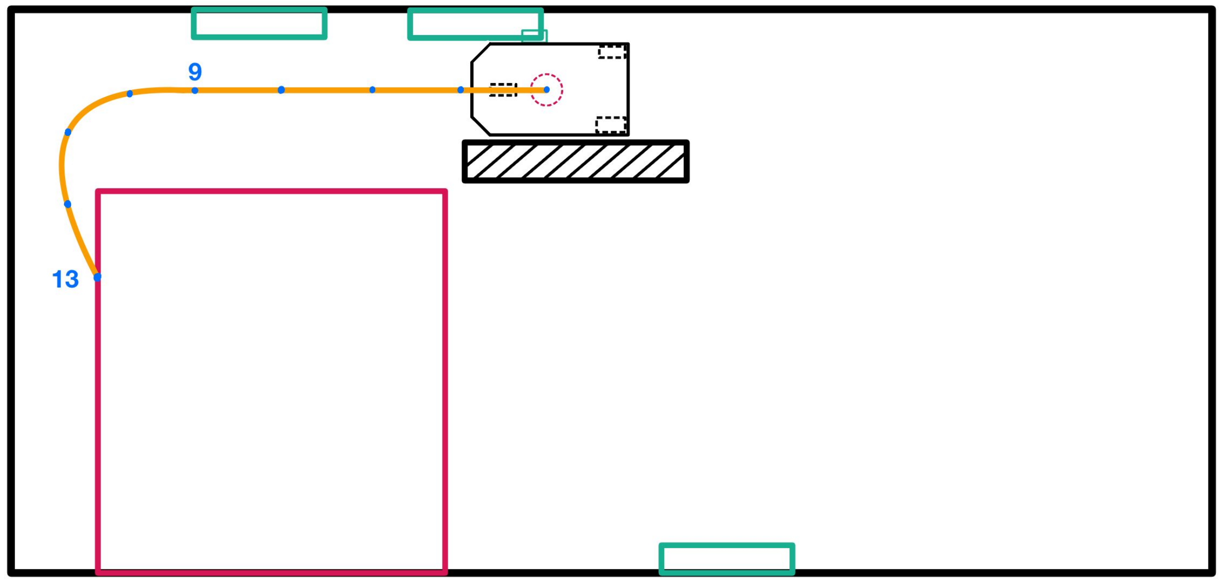


Dynamic Mode Array

- Mode: Disinfection
- Mode End Index: 9
- DSX Spray State: 1
- CH State: 1

.
.
.

- Mode: stop
- Mode End Index: 13
- DSX Spray State: 0
- CH State: -2

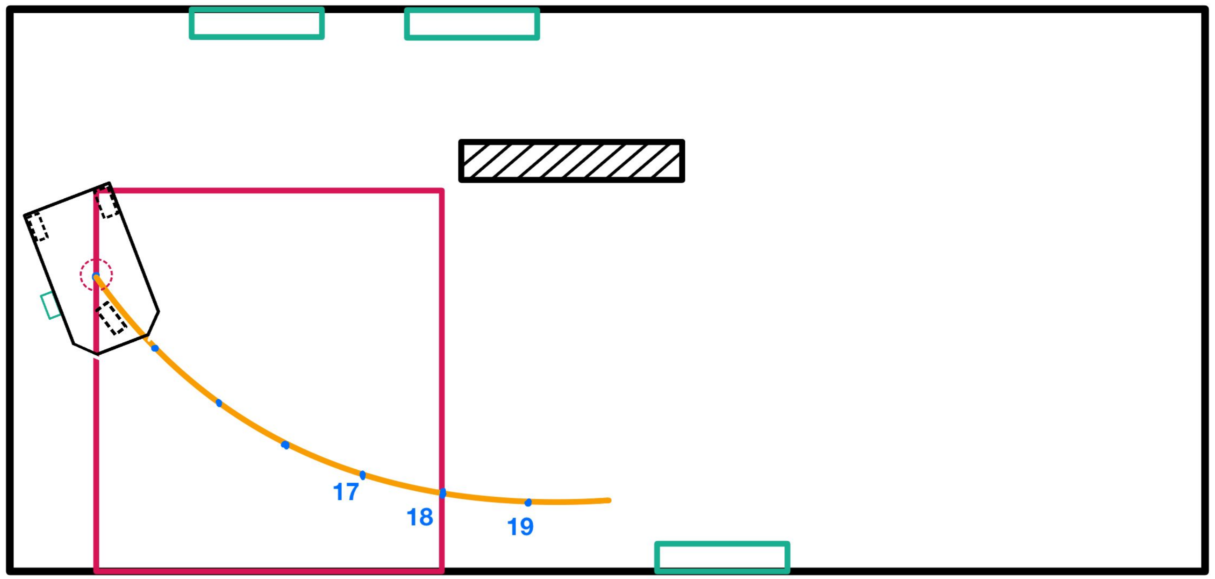


Dynamic Mode Array

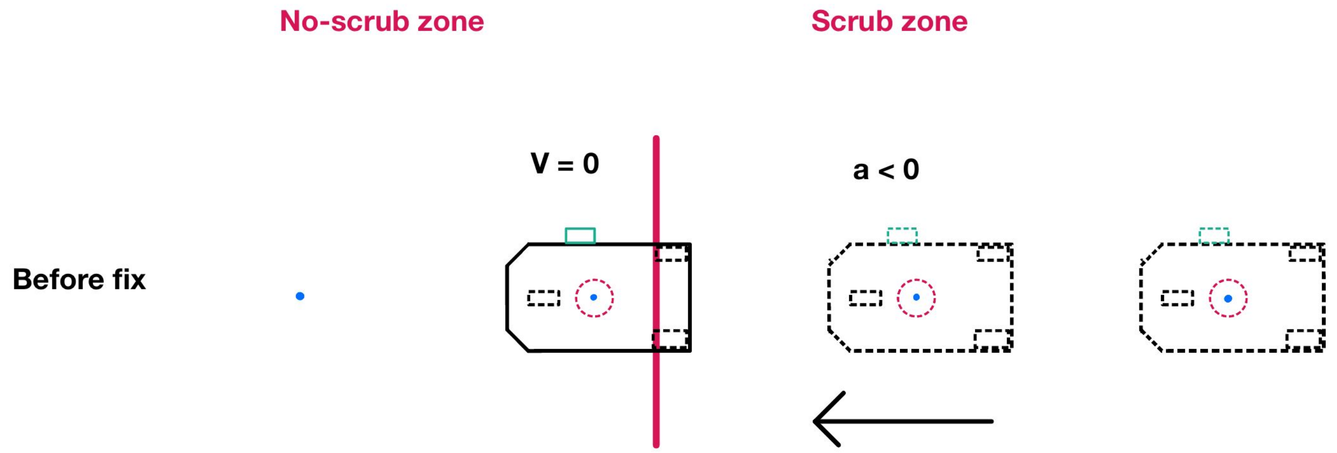
- Mode: no scrub
- Mode End Index: 16
- DSX Spray State: 0
- CH State: 0

.
.
.

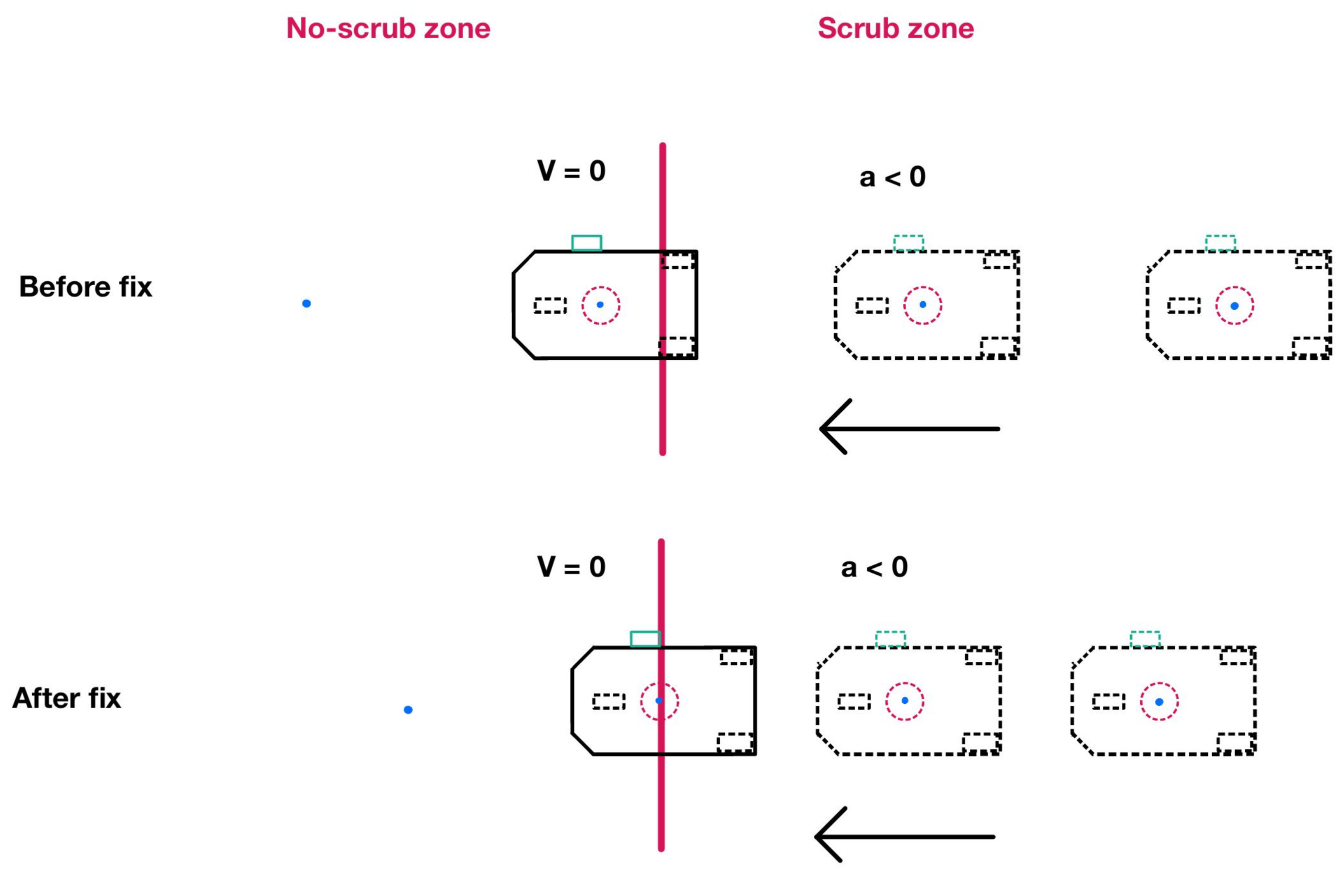
- Mode: stop
- Mode End Index: 18
- DSX Spray State: 0
- CH State: -1



Scenario: Momentary Stop

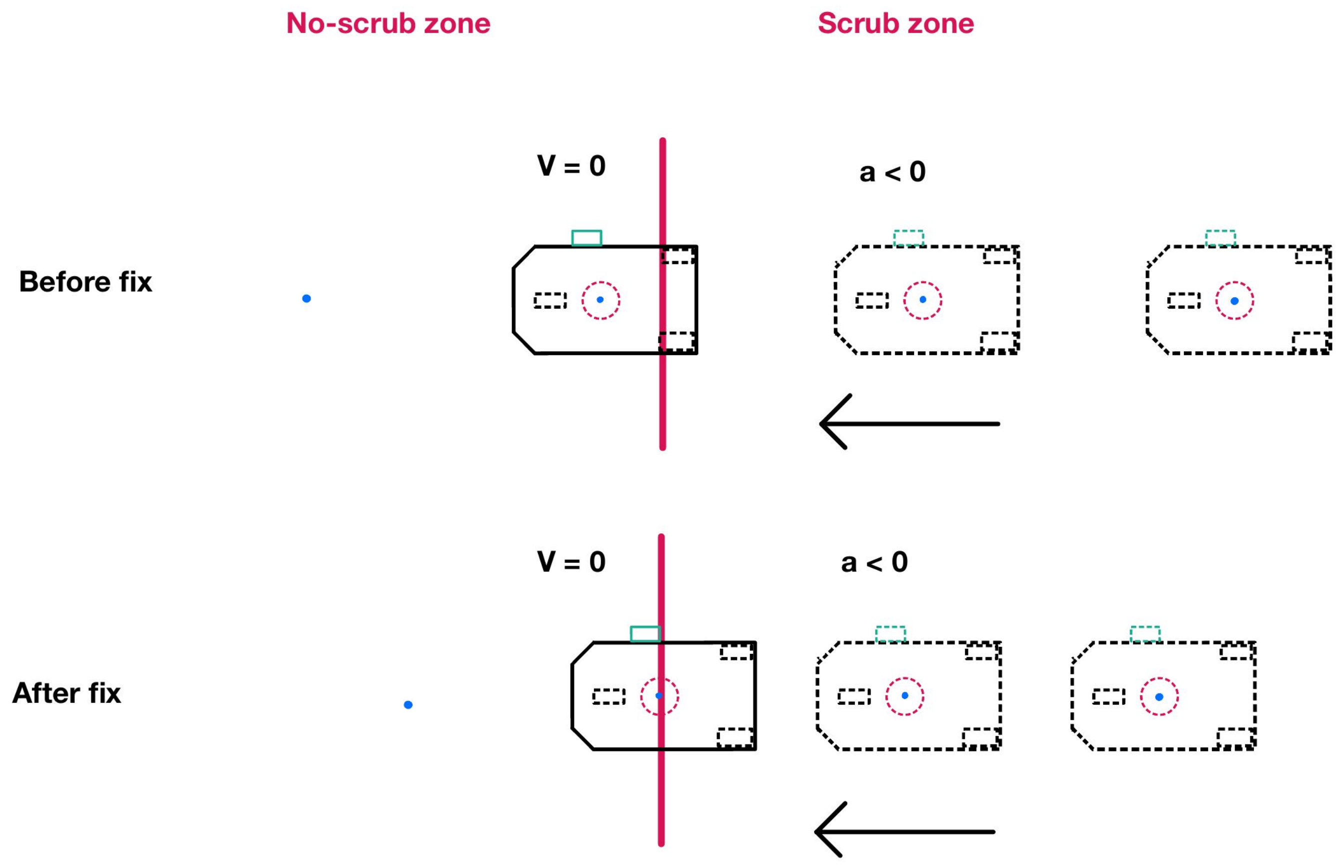


Scenario: Momentary Stop

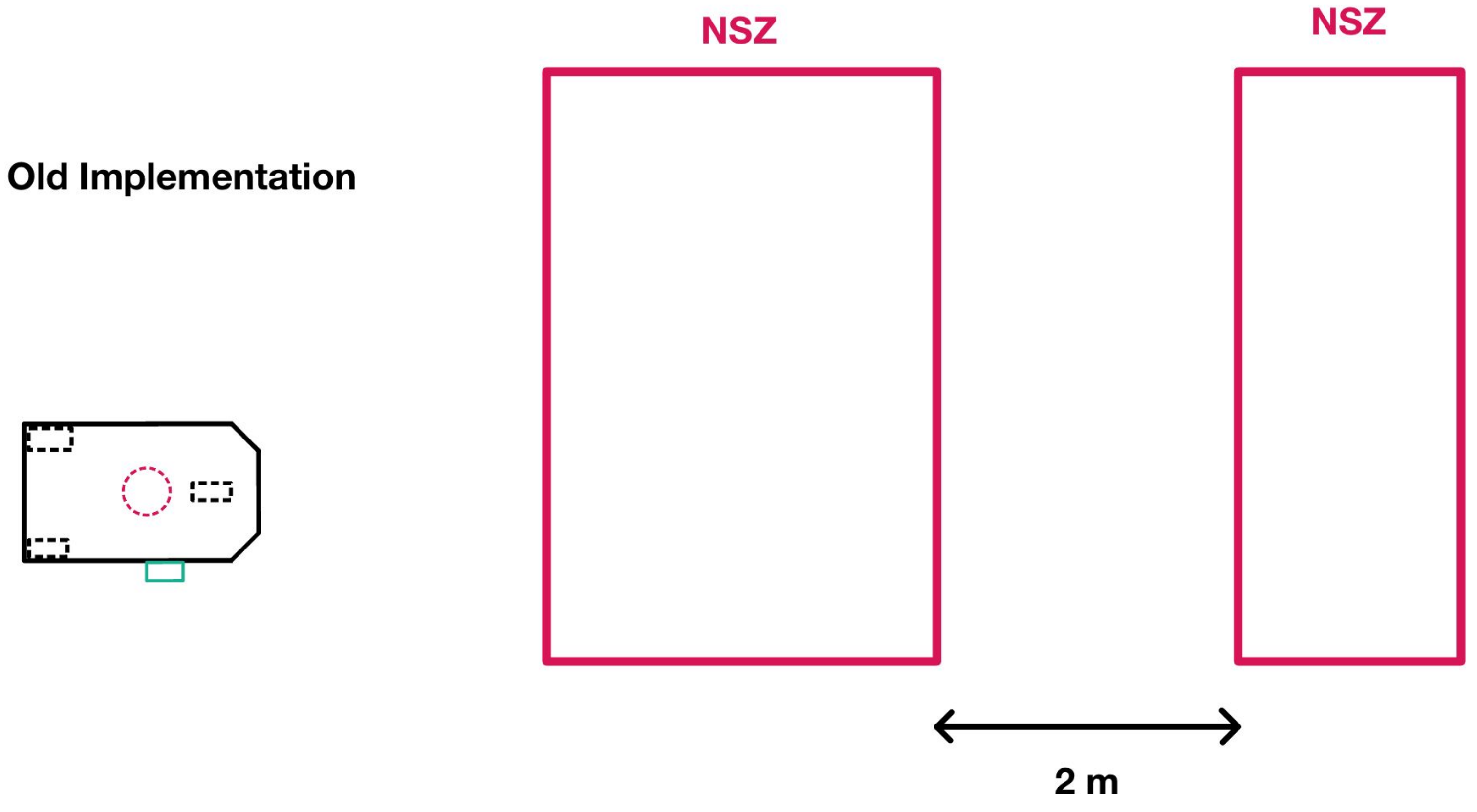


Scenario: Momentary Stop

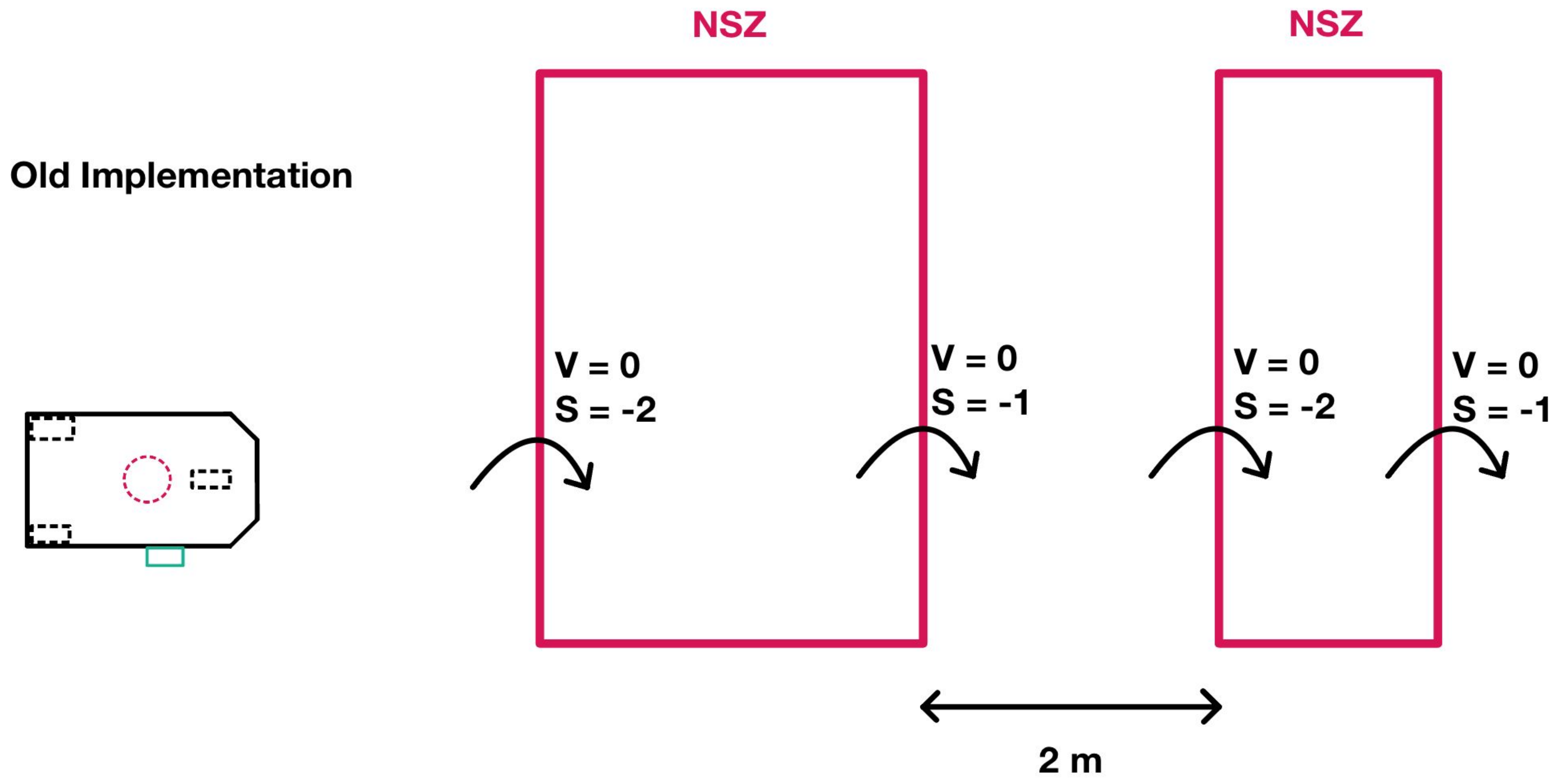
~ 5% improvement in accuracy of average cleaned area



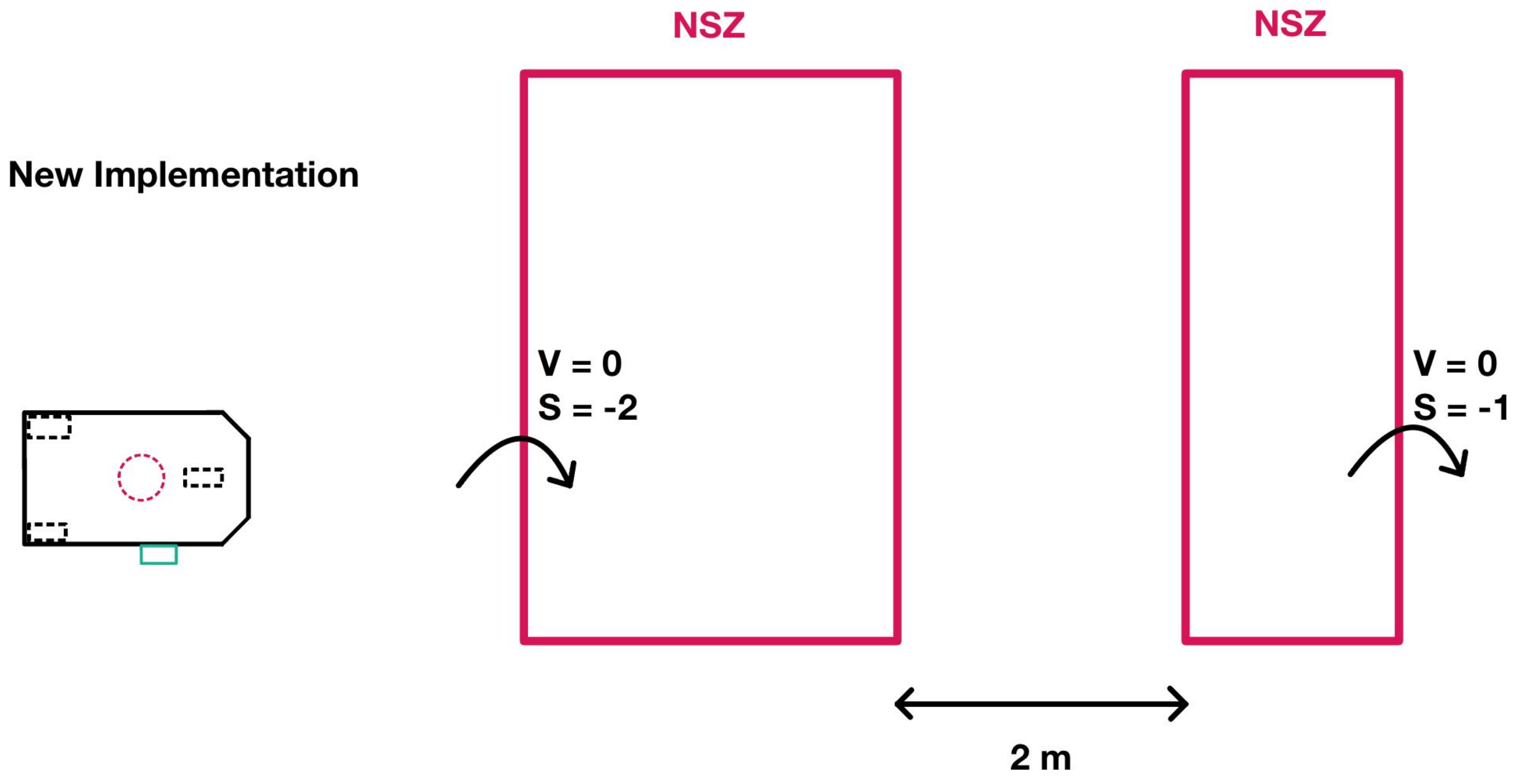
Scenario: Temporary Exit



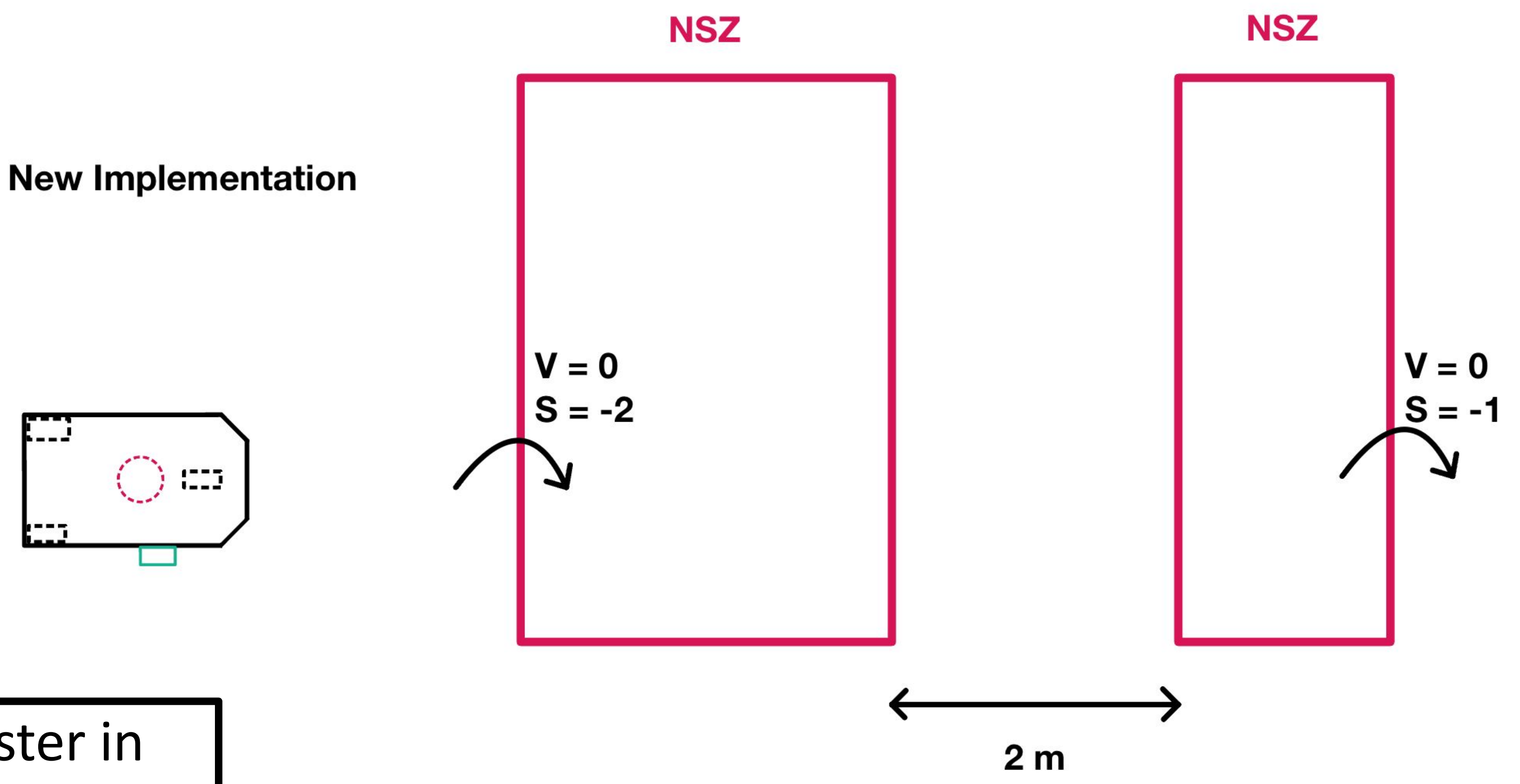
Scenario: Temporary Exit



Scenario: Temporary Exit

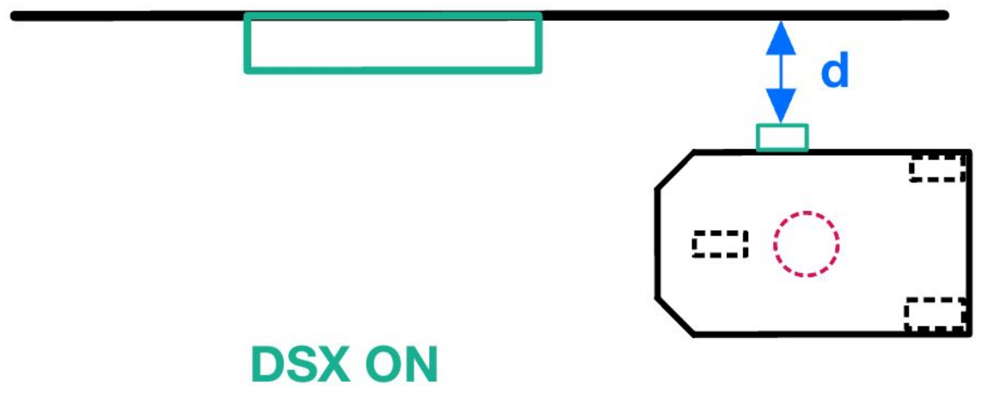


Scenario: Temporary Exit

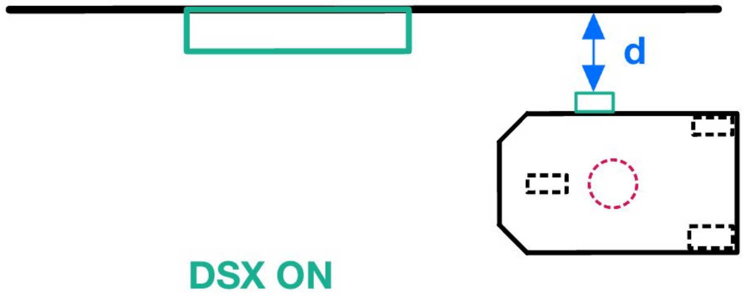


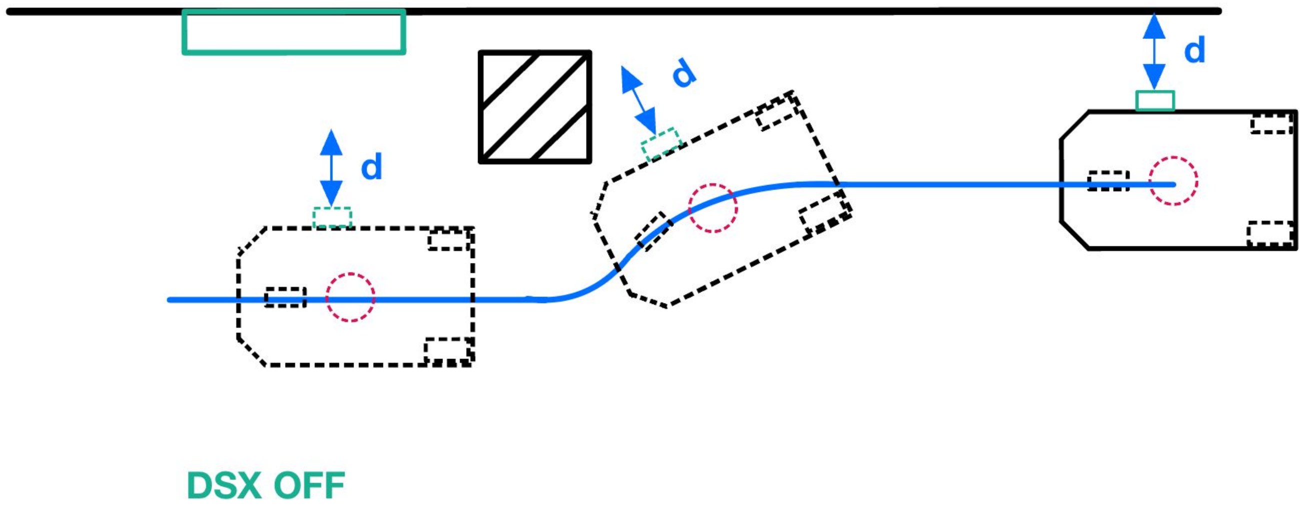
~**10%** faster in total cleaning time

Distance Clearance for Disinfection



Scenario: Obstacle Obstruction





Lessons I Learned

- Comprehensive simulation testing
- In-depth knowledge of Local Planner and Local Controller
- ROS Services

Conclusion

- New node to keep the same functionalities for Local Planner and Local Controller and most of Hardware Controller
- New implementation allows for:
  - New, efficient behaviour in Hardware Controller
  - Design for usability of code for new hardware elements
  - Easier debugging of robot behaviour

Thanks for Watching!

Q&A