# *Safety rules* should help the robot succeed instead of hindering it!

## A Domain-Specific Language to express dynamic robot safety rules

**Momina Rizwan**, Christoph Reichenbach and Volker Krueger

### *Motivation & Research Goal*

Ensuring robot functional safety is challenging in a dynamic and unpredictable environment.

"A functionally safe autonomous system performs correctly given a set of inputs and if it can't, it fails predictably."

Completely turning off the robot whenever a robot senses a human crossing its path, hinders it from succeeding. In our research, we replicate the work by [S.Adam et. al.] [1] and add dynamic safety rules to adapt to such changing environment.

### *Safety scenarios*

| Scenario | Static Behavior | Dynamic Behavior |
|---|---|---|
| **Ramps** | Stop the robot | Speed up the robot a bit so it can cross a gentle slope. If the slope is steep, slow down, go back and find a new path. |
| **Unlocked partially closed door** | Stop the robot | If the map indicates that an obstacle is a pushable door, try to push it with a low speed to check if it can be opened. |
| **Robot arm senses an object** | Stop the robot | Instead of stopping the arm, go into gravity-free mode first and continue performing the task as soon as the obstacle goes away. |

### *Safety Node As a Filter Node in ROS*

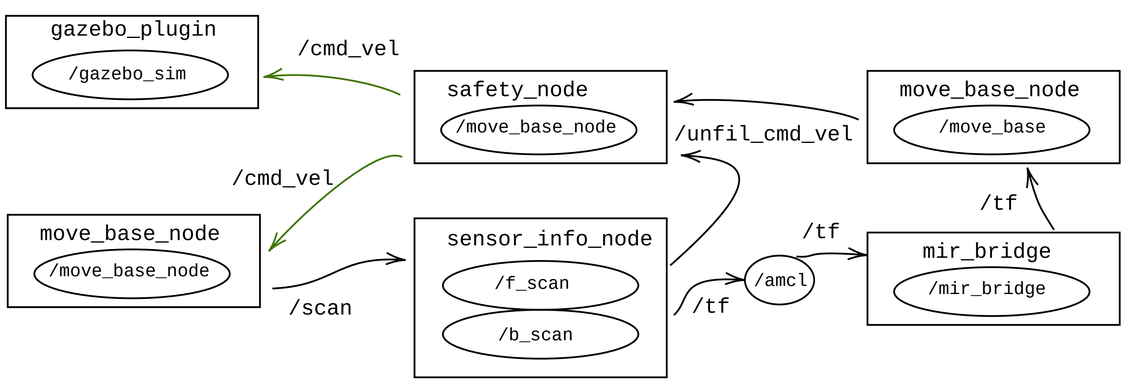


Fig. 1: ROS publisher/subscriber nodes and topics shown as a graph.

### *Why use a DSL?*

"concise syntax aimed at robot users, not software engineers [2]"

- Allows static analysis to report bugs earlier.
- Effective error reporting.

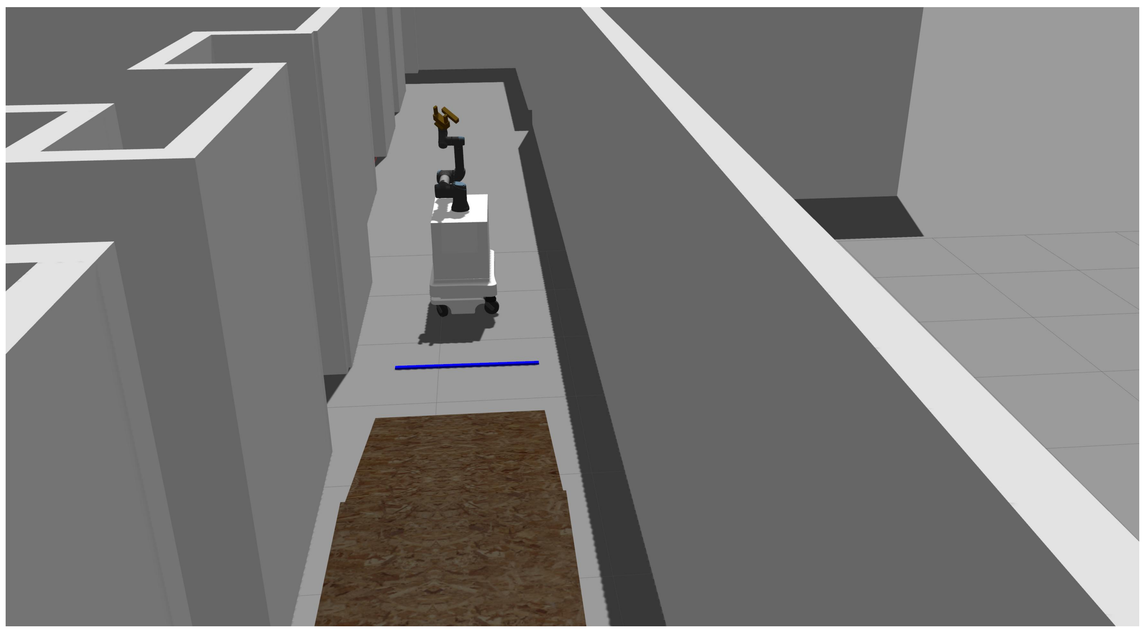

Fig. 2: An uneven terrain and ramps simulated in Gazebo simulator using ROS.

### *A code snippet*

```
action moveBack;
action lowSpeed;
action increaseSpeed;
const maxSpeed = 0.5 m/s
const reasonabletilt = 10 deg
const maxtilt = 20 deg
input orientation = topic imuInformation
entity imuSensorSystem
{
  gentleSlope :
    orientation.pitch() not in reasonabletilt
    for 4.0 sec;
  steepRamp :
    orientation.pitch() not in maxtilt
    for 4.0 sec;
}
entity driveSystem {
  maxspeedExceeded :
    linearSpeed > maxSpeed for 2.0 sec ;
}
if imuSensorsystem.gentleSlope and driveSystem.moving
then { increaseSpeed;};
if imuSensorsystem.steepRamp and driveSystem.moving
then { lowSpeed; moveBack; };
```

Fig. 3: Syntax in the style of [Adam et al.][1] *

### References

[1] Sorin Adam, Morten Larsen, Kjeld Jensen, and Ulrik Pagh Schultz. Rule-based dynamic safety monitoring for mobile robots. *Journal of Software Engineering for Robotics*, 7(1):121–141, 2016.

[2] Arne Nordmann, Nico Hochgeschwender, and Sebastian Wrede. A survey on domain-specific languages in robotics. In *International conference on simulation, modeling, and programming for autonomous robots*, pages 195–206. Springer, 2014.

WASP | WALLENBERG AI, AUTONOMOUS SYSTEMS AND SOFTWARE PROGRAM