rapyuta robotics

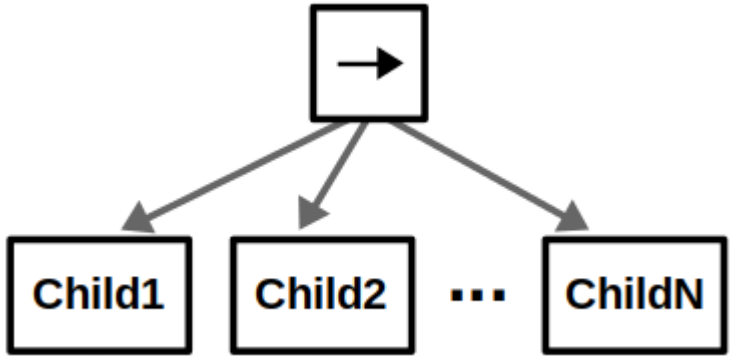
Navigation “à la carte”
profile and strategy as you go

Jorge Santos Simon

2024-10-23

Presentation outline

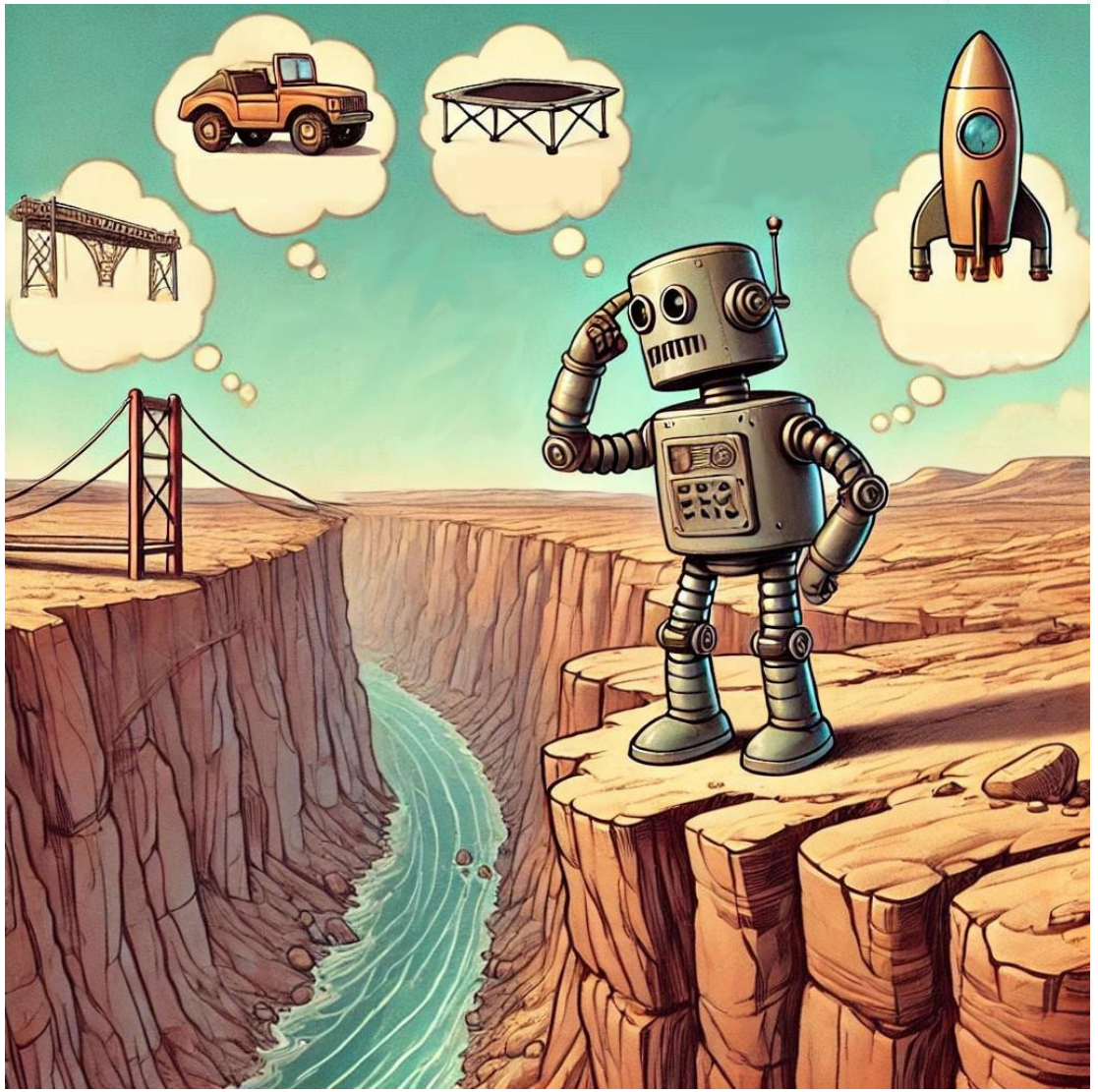
- Motivation
- A bit of history
- What's a profile?
- Trigger mechanisms
- How does it work?
- Example profiles
- Put it to work
- "We are hiring" time



Motivation

a fit all navigation?

not in Rapyuta Robotics...



Motivation

At Rapyuta Robotics, we work with different robots and environments

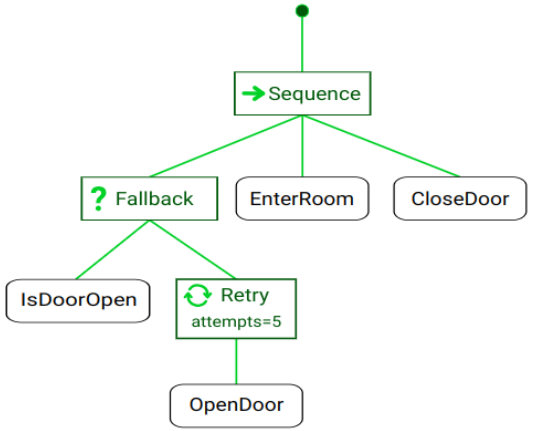
A bit of history...

We developed different navigation strategies, all encoded as Behavior Trees (BT from now)

Still, that was not enough for particular cases, like picking a pallet, or docking to a charging station

So we created temporal configurations that apply only during the current navigation

Put all together…



+

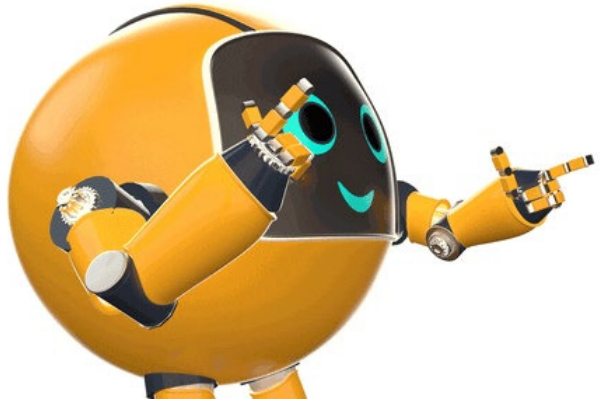
```
nav_server:
  planner: HybridAStar
  controller: PathFollower
move_base_flex:
  oscillation_distance: 0.1
  PathFollower:
    max_vel_trans: 0.3
    min_lookahead_dist: 0.2
    speed_inflation_weight: 0.0
```

+

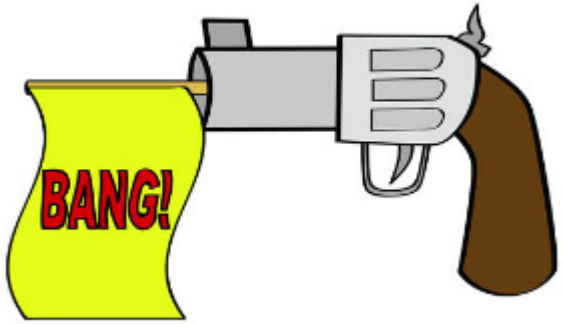
+

=

navigation profiles!



trigger mechanisms



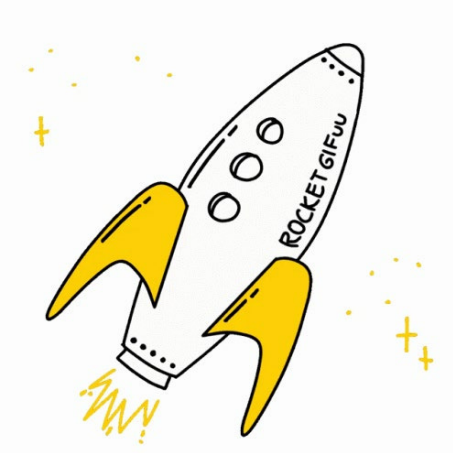
+

But what's a navigation profile?

it's... well, just a yaml file

that specifies:

- the strategy (i.e. BT)
- [planner, controller]
- [goal tolerances]
- [multi-node configuration]

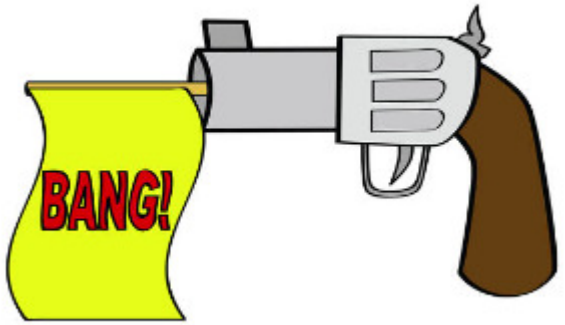



```yaml
strategy: graph_free
tolerance:
  linear: 0.1
  angular: 3.15
configuration:
  nav_server:
    planner: HybridAStar
    controller: PathFollower
  move_base_flex:
    oscillation_distance: 0.1

    PathFollower:
      # make path follower slower more prudent
      max_vel_trans: 0.3
      min_lookahead_dist: 0.2

      # allow it to go very close to obstacles
      speed_inflation_weight: 0.0
      obstacle_approach_distance: 0.0
```

Trigger mechanisms

Profiles can be…

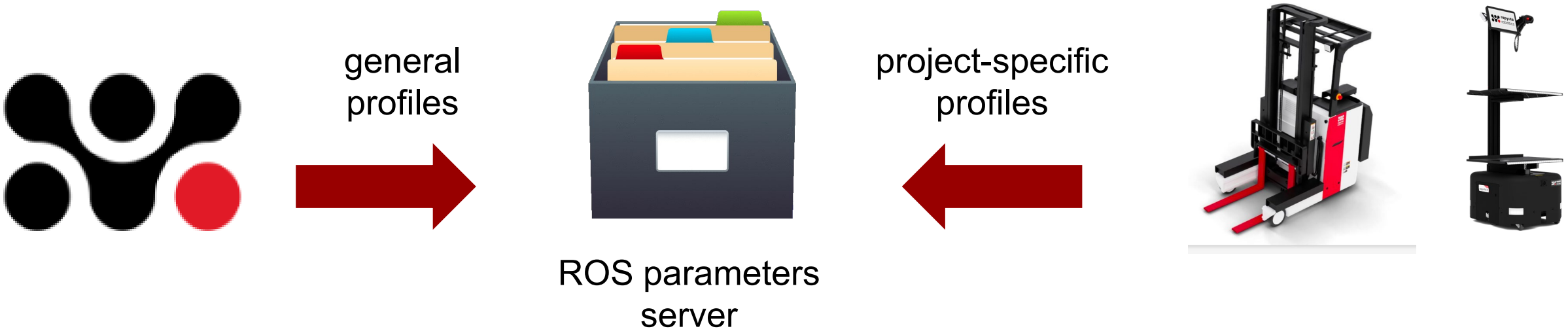
- selected with the navigation goal
- applied in particular zones
- applied during particular times
- applied under particular conditions
- changed by the operator



all at runtime, without stopping the robot!

How does it work?

- A script loads the navigation profiles at launch time
  - depending on the application, we can also load project-specific profiles
  - all are loaded in a common namespace on the ROS parameter server



How does it work?

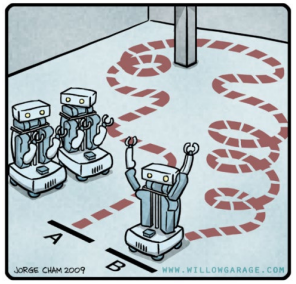
- Upon receiving a new goal, the navigation action server (NAS for friends) loads the strategy BT and configuration
  - both are kept loaded for further usage

How does it work?

Navigation starts!

NAS applies the configuration and starts ticking the BT

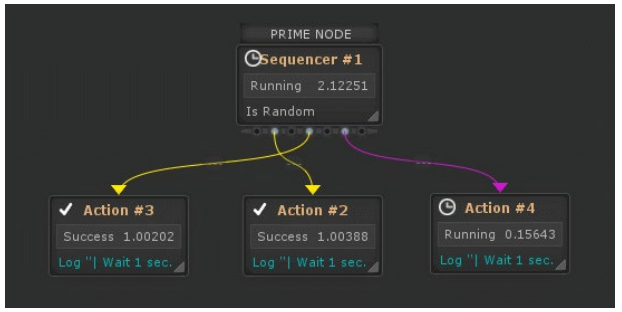
- it cancels the previous goal if the strategy changes
- preempts it otherwise



MBF
(and other nodes)

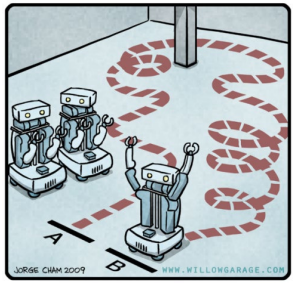
reconfigure

run BT



How does it work?

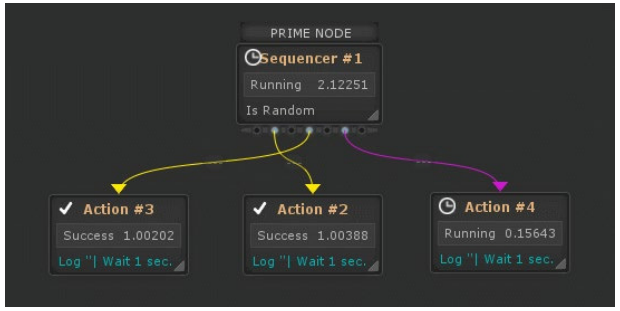
Upon completion, all the configuration changes are reverted



MBF
(and other nodes)

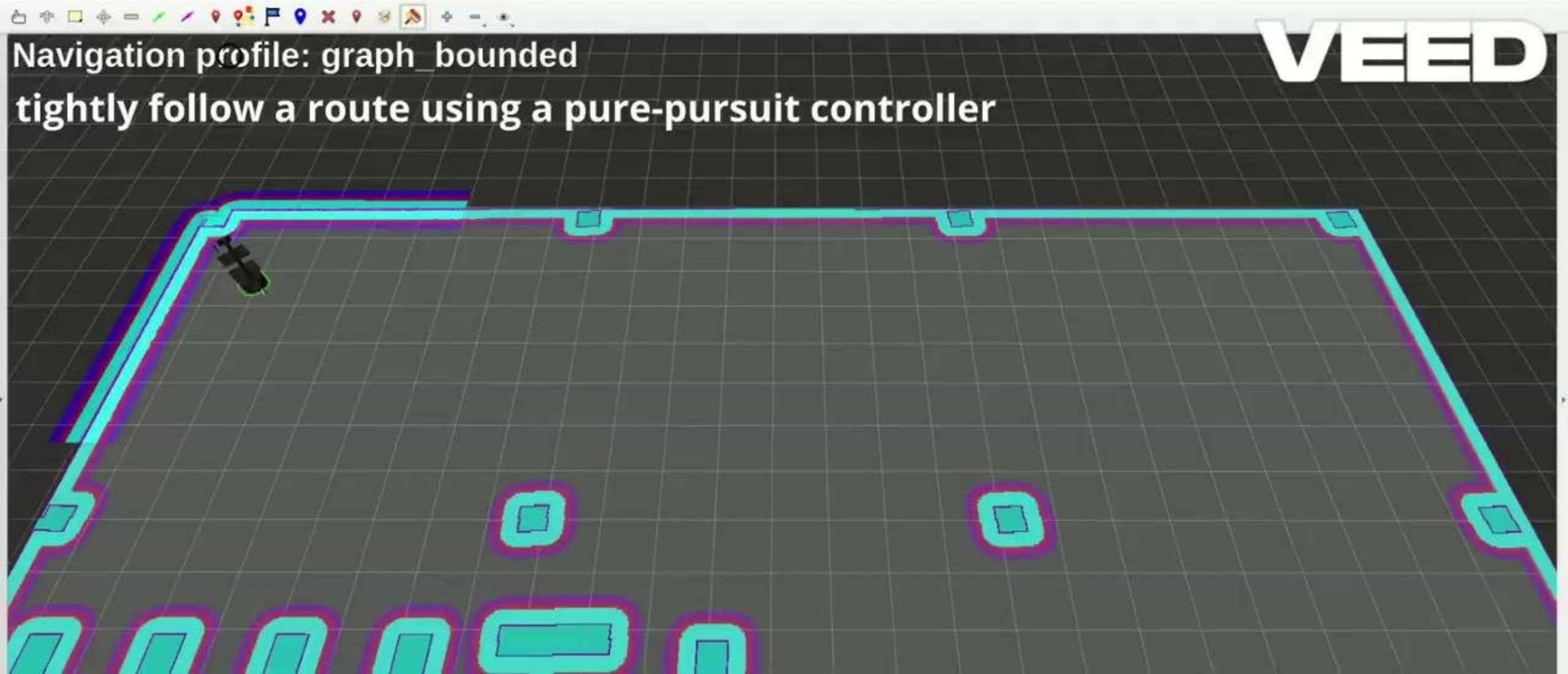
revert changes

finished



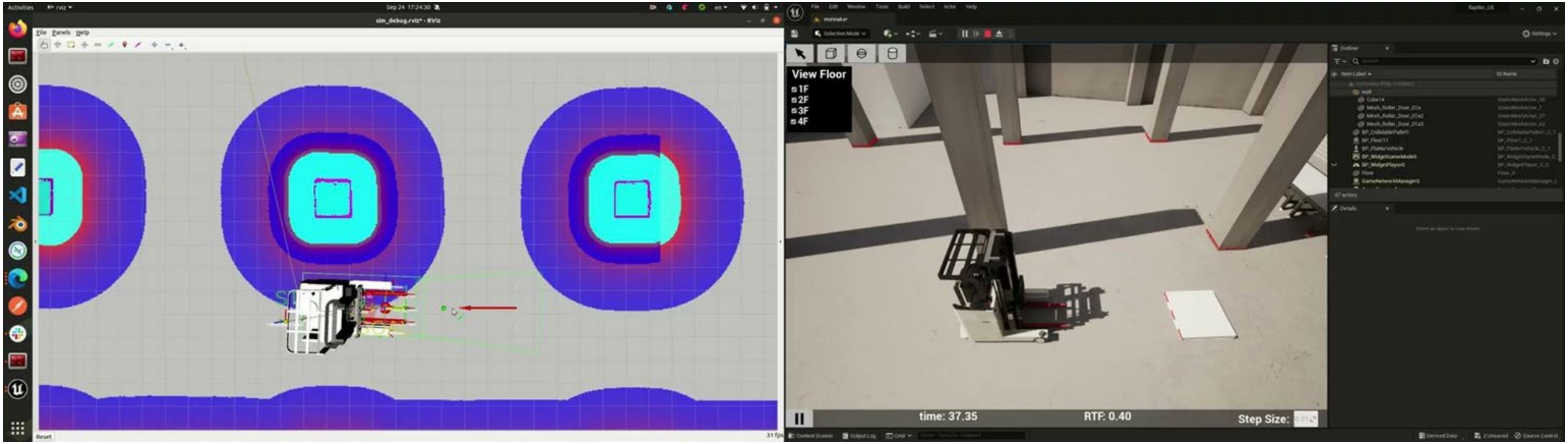
Example profiles

Graph Bounded / Guided / Free, switching as you go



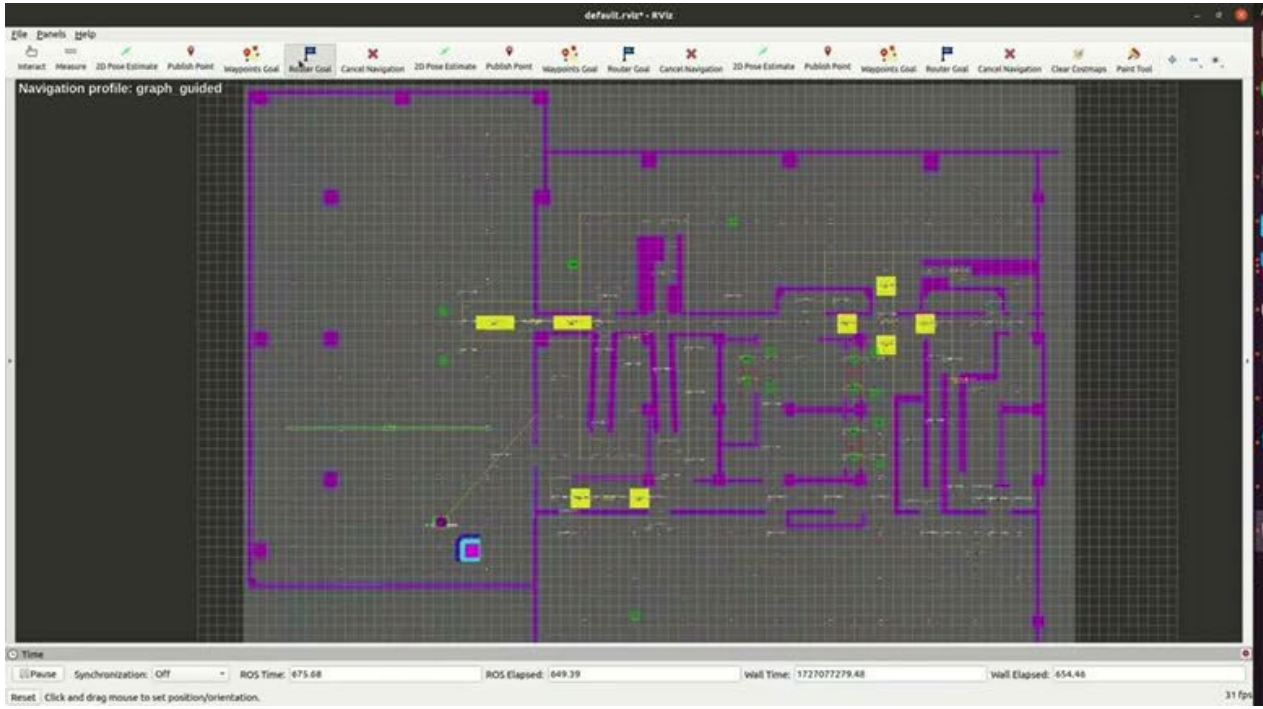
Example profiles

- Pick, drop, charge…
  i.e. maneuvers that typically require to move slowly and mostly straight



Use cases

- Switch to graph bounded on difficult passages
- Switch to graph free to run for your (robotic) life



DIY

- Not open source, but…
  - BehaviorTree.CPP
  - move_base_flex
  - Nav2 (can load different BTs)
- TODO
  - multi-node reconfigure client
  - trigger mechanisms

We're hiring !

Actively Hiring Positions

- Robotics Software Engineer
- Frontend Engineer
- Software Engineer – Distributed Intelligence
- Electrical Engineer AMR / AFL / ASRS

Other

- Hiring Page
- Inquiries here :
  hiring-tyo_sig@rapyuta-robotics.com

Thank you!

ありがとう!

time for questions