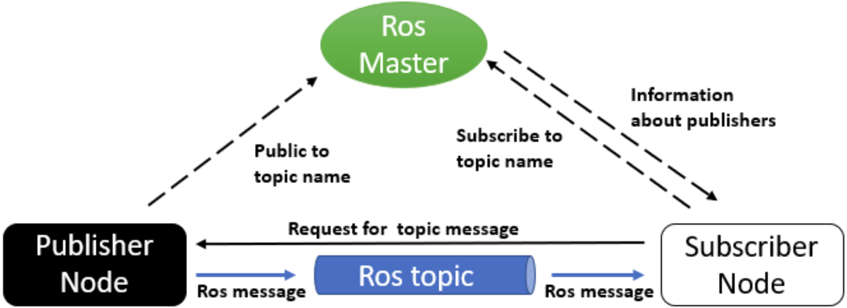
Introduction to Nav2

In Robotics, before 2007

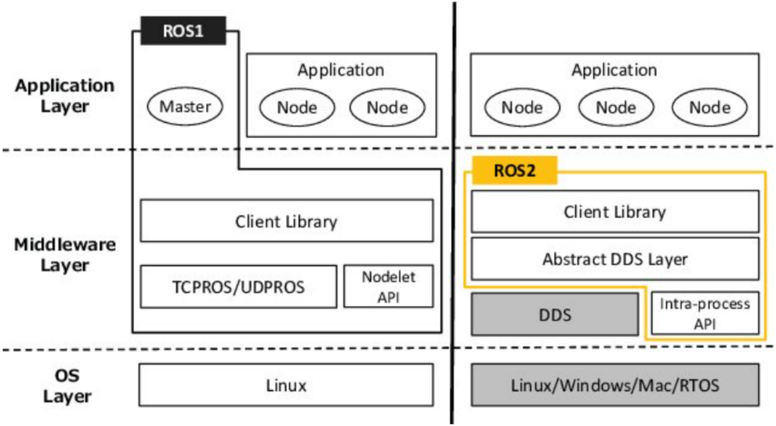
1. Problem of reusability
2. Which standard to follow
3. Multiple drivers for the same device, write internal communications, e.g., ICP (Inter-Process Communication), managing of shared data
4. Implements the same algorithm in different standards
5. Have to start from scratch most of the time

ROS1 Big Picture



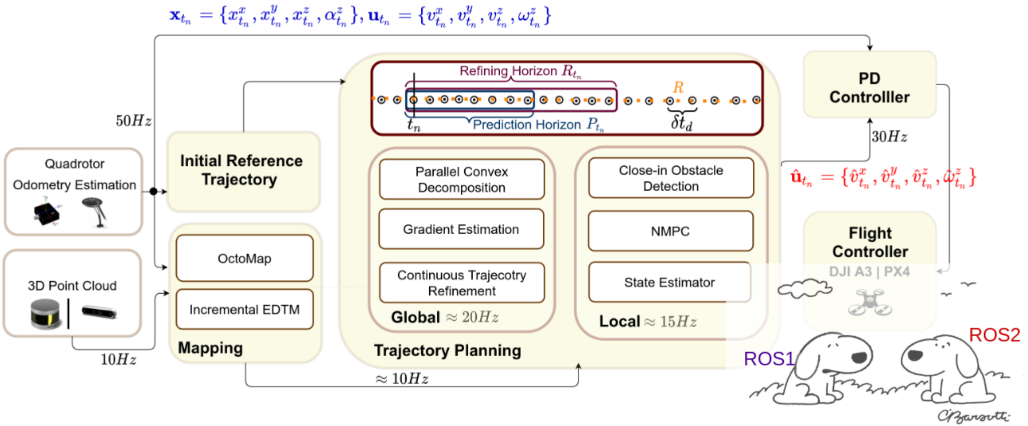
@https://trojrobert.github.io/hands-on-introdution-to-robot-operating-system(ros)/

ROS2 Big Picture



Y. Maruyama, S. Kato and T. Azumi, "Exploring the performance of ROS2," 2016 International Conference on Embedded Software (EMSOFT), Pittsburgh, PA, USA, 2016, pp. 1-10, doi: 10.1145/2968478.2968502.

Why do we need Nav2?



$\mathbf{x}_{t_n} = \{x^x_{t_n}, x^y_{t_n}, x^z_{t_n}, \alpha^z_{t_n}\}, \mathbf{u}_{t_n} = \{v^x_{t_n}, v^y_{t_n}, v^z_{t_n}, \omega^z_{t_n}\}$

$50Hz$

Quadrotor Odometry Estimation

3D Point Cloud

$10Hz$

Initial Reference Trajectory

OctoMap

Incremental EDTM

Mapping

$\approx 10Hz$

Refining Horizon R_{t_n}

R

t_n

Prediction Horizon P_{t_n}

δt_d

Parallel Convex Decomposition

Gradient Estimation

Continuous Trajecotry Refinement

Global $\approx 20Hz$

Close-in Obstacle Detection

NMPC

State Estimator

Local $\approx 15Hz$

Trajectory Planning

PD Controlller

$30Hz$

$\hat{\mathbf{u}}_{t_n} = \{\hat{v}^x_{t_n}, \hat{v}^y_{t_n}, \hat{v}^z_{t_n}, \hat{\omega}^z_{t_n}\}$

Flight Controller

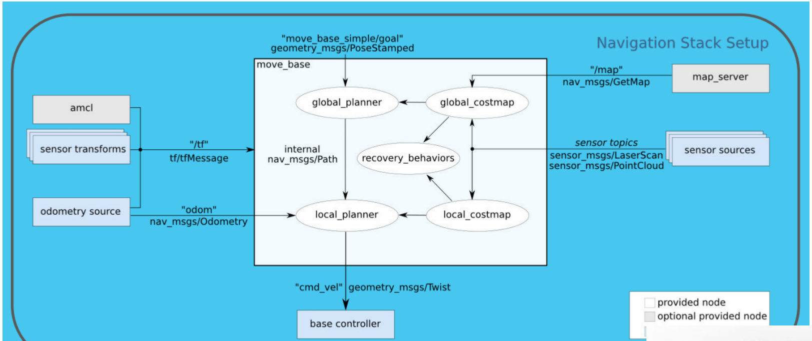
DJI A3 | PX4

ROS1

ROS2

Kulathunga, G., Hamed, H., Devitt, D., Klimchik, A. (2022). Optimization-based trajectory tracking approach for multi-rotor aerial vehicles in unknown environments. IEEE Robotics and Automation Letters, 7(2), 4598-4605.

As a ROS1 developer?

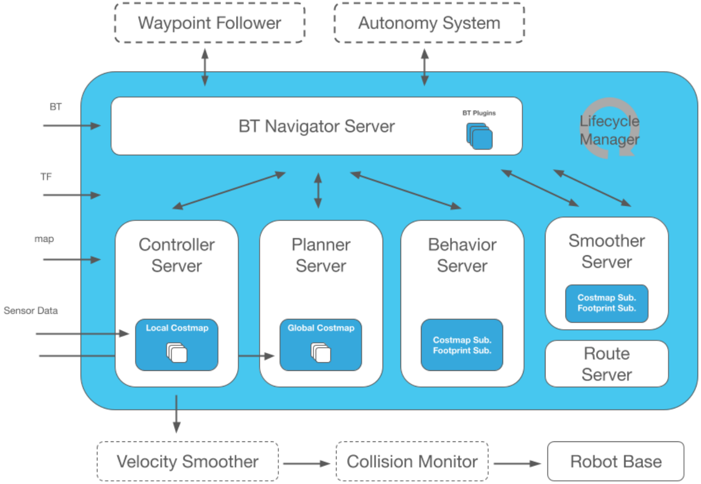


As a ROS2 developer?

The motivation for Nav2

- **Addresses many of the shortcomings of the ROS 1**: modularity and maintainability
- **Product-ready middleware**
- **Quality and reliability focused**
- **Embedded and real-time support**
- **Multi-agent support** can be used to coordinate the navigation of multiple robots in a coordinated and efficient manner
- **Best security practices** includes a number of security features, such as authentication and authorization, to protect it from unauthorized access and modification
  https://docs.ros.org/en/rolling/Concepts/Intermediate/About-Security.html

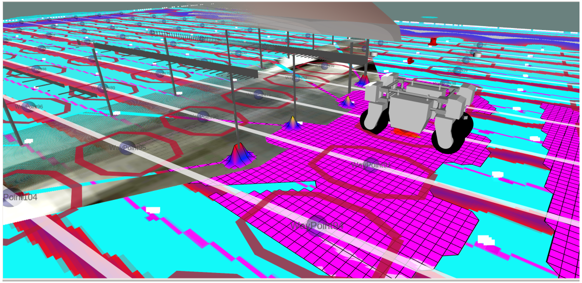
Nav2 Overview



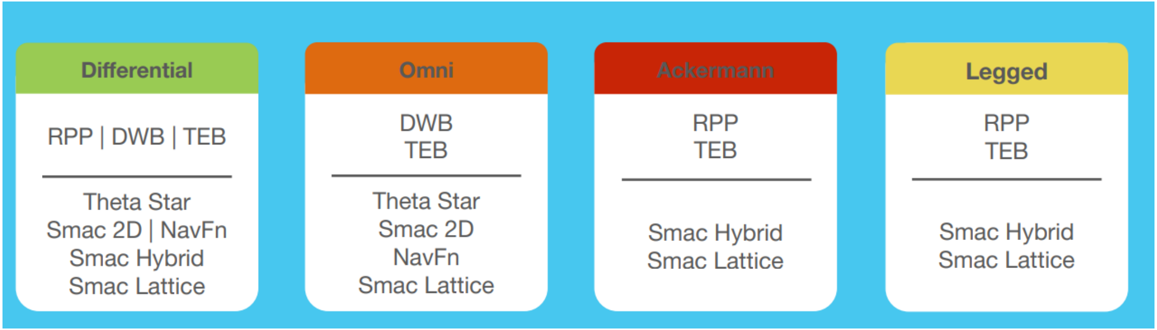
https://navigation.ros.org/

Migrated to ROS2

- Costmap 2D
- AMCL / SLAM Toolbox with some improvement
- NavFn Planner
- Map Server



New Planning Plugins

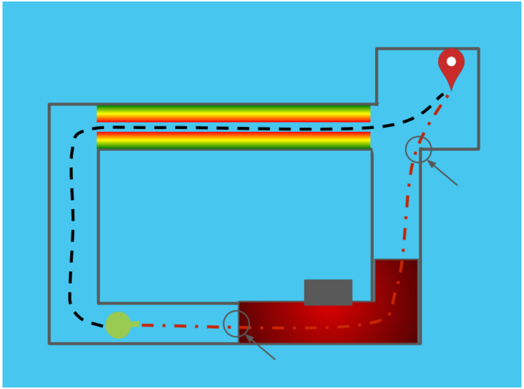


Differential	Omni	Ackermann	Legged
RPP | DWB | TEB	DWB TEB	RPP TEB	RPP TEB
Theta Star Smac 2D | NavFn Smac Hybrid Smac Lattice	Theta Star Smac 2D NavFn Smac Lattice	Smac Hybrid Smac Lattice	Smac Hybrid Smac Lattice

https://roscon.ros.org/jp/2021/presentations/8.pdf

Smac Planner

- Cost-Aware Obstacle Heuristic (Steers towards the solution, away from obstacles, Uses cost, not just binary obstacles)
- Search Penalty Functions (Reverse, Change Direction, Non-Straight, Cost)
- Finds exact and feasible paths to the goal



http://download.ros.org/downloads/roscon/2022/OnUseofNav2SmacPlanners.pdf

Behaviours in Robotics

- Reactive behaviour occurs when the **action** is related only to **the occurrence of an event** and does not depend **on data stored in memory (state).** [1]

[1]. Ben-Ari, M., Mondada, F. (2018). Reactive Behavior. In: Elements of Robotics. Springer, Cham. https://doi.org/10.1007/978-3-319-62533-1_3
[2]. Colledanchise, M., Ögren, P. (2018). Behavior trees in robotics and AI: An introduction. CRC Press.

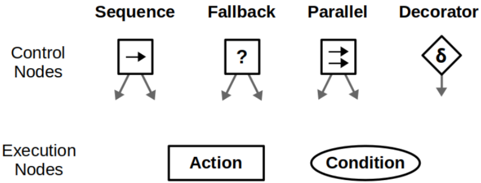
Behaviours in Robotics

- Reactive behaviour occurs when the **action** is related only to **the occurrence of an event** and does not depend **on data stored in memory (state).** [1]
- Finite State Machine

[1]. Ben-Ari, M., Mondada, F. (2018). Reactive Behavior. In: Elements of Robotics. Springer, Cham. https://doi.org/10.1007/978-3-319-62533-1_3
[2]. Colledanchise, M., Ögren, P. (2018). Behavior trees in robotics and AI: An introduction. CRC Press.

Behaviours in Robotics

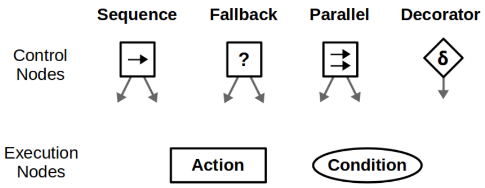
- Reactive behaviour occurs when the **action** is related only to **the occurrence of an event** and does not depend **on data stored in memory (state).** [1]
- Finite State Machine
- Behavior trees



[1]. Ben-Ari, M., Mondada, F. (2018). Reactive Behavior. In: Elements of Robotics. Springer, Cham. https://doi.org/10.1007/978-3-319-62533-1_3
[2]. Colledanchise, M., Ögren, P. (2018). Behavior trees in robotics and AI: An introduction. CRC Press.

Behaviours in Robotics

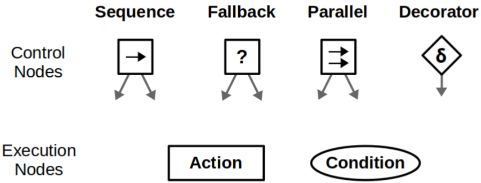
- Reactive behaviour occurs when the **action** is related only to **the occurrence of an event** and does not depend **on data stored in memory (state).** [1]
- Finite State Machine
- Behavior trees
  - **Leaf nodes** can be executed in one of these states: success, failure, or running. **Internal nodes** control tree traversal [2]



[1]. Ben-Ari, M., Mondada, F. (2018). Reactive Behavior. In: Elements of Robotics. Springer, Cham. https://doi.org/10.1007/978-3-319-62533-1_3
[2]. Colledanchise, M., Ögren, P. (2018). Behavior trees in robotics and AI: An introduction. CRC Press.

Behaviours in Robotics

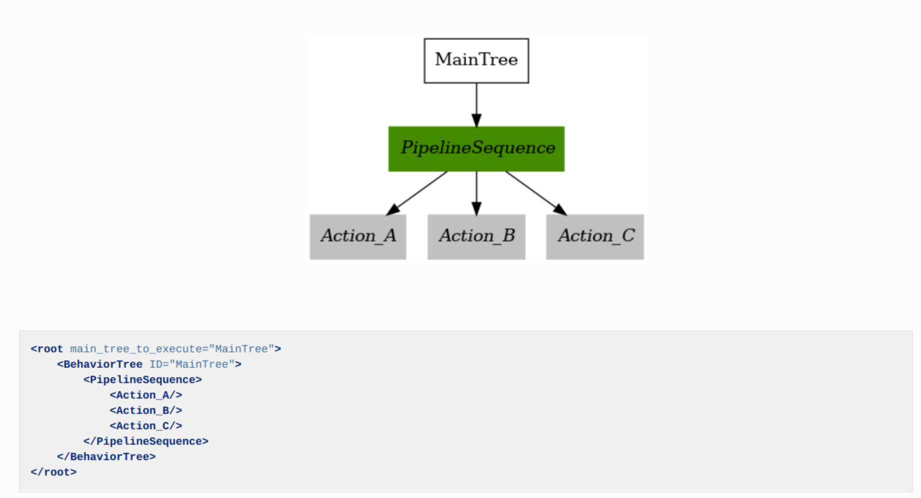
- Reactive behaviour occurs when the **action** is related only to **the occurrence of an event** and does not depend **on data stored in memory (state).** [1]
- Finite State Machine
- Behavior trees
  - **Leaf nodes** can be executed in one of these states: success, failure, or running. **Internal nodes** control tree traversal [2]



- BehaviorTree.CPP 4.4 https://www.behaviortree.dev/

[1]. Ben-Ari, M., Mondada, F. (2018). Reactive Behavior. In: Elements of Robotics. Springer, Cham. https://doi.org/10.1007/978-3-319-62533-1_3
[2]. Colledanchise, M., Ögren, P. (2018). Behavior trees in robotics and AI: An introduction. CRC Press.

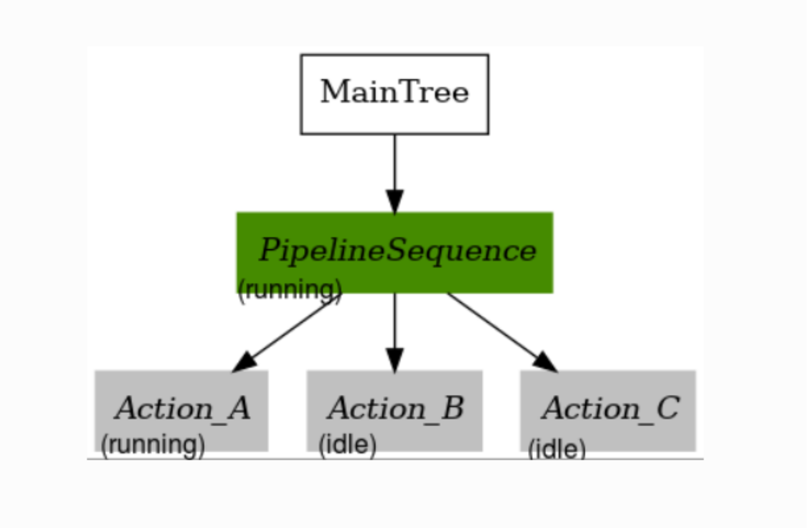
Nav2 Behavior Trees



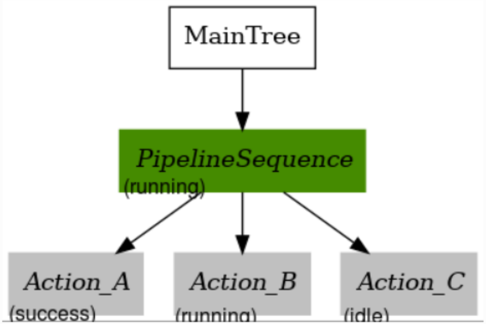
```
<root main_tree_to_execute="MainTree">
    <BehaviorTree ID="MainTree">
        <PipelineSequence>
            <Action_A/>
            <Action_B/>
            <Action_C/>
        </PipelineSequence>
    </BehaviorTree>
</root>
```

https://navigation.ros.org/plugin_tutorials/docs/writing_new_bt_plugin.html

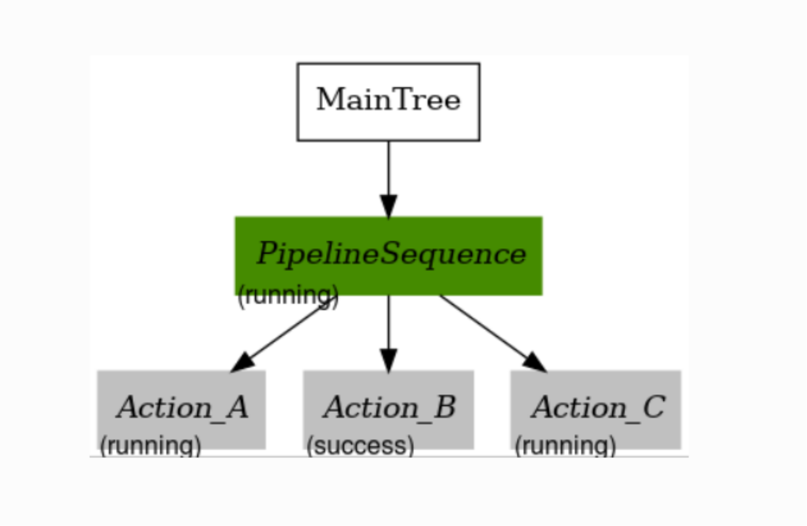
Control: PipelineSequence



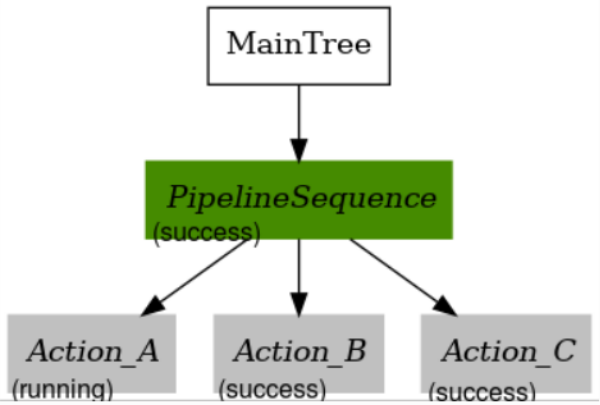
Control: PipelineSequence



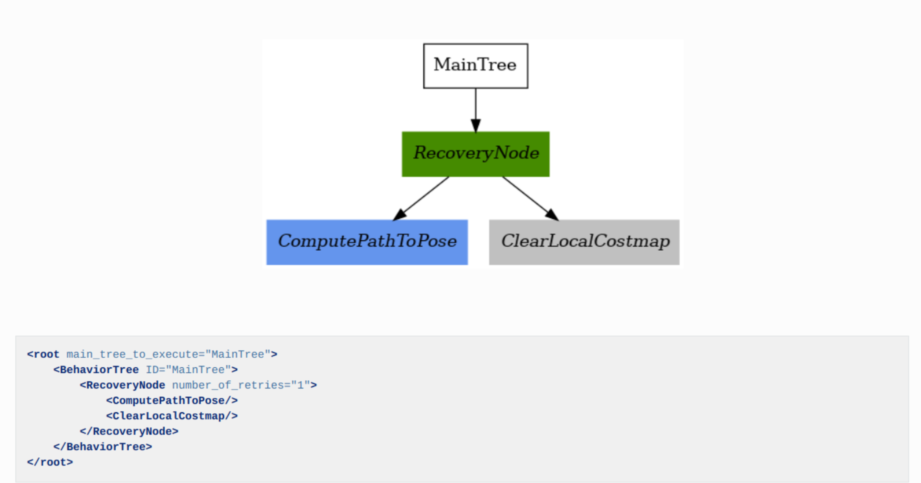
Control: PipelineSequence



Control: PipelineSequence



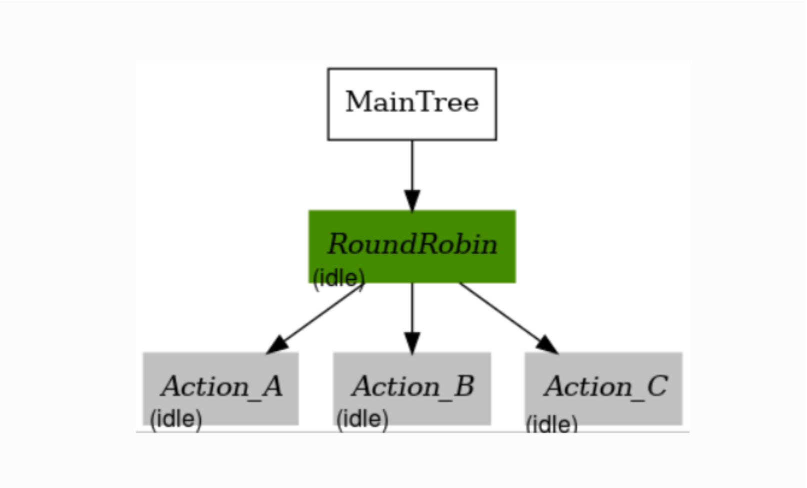
Control: Recovery



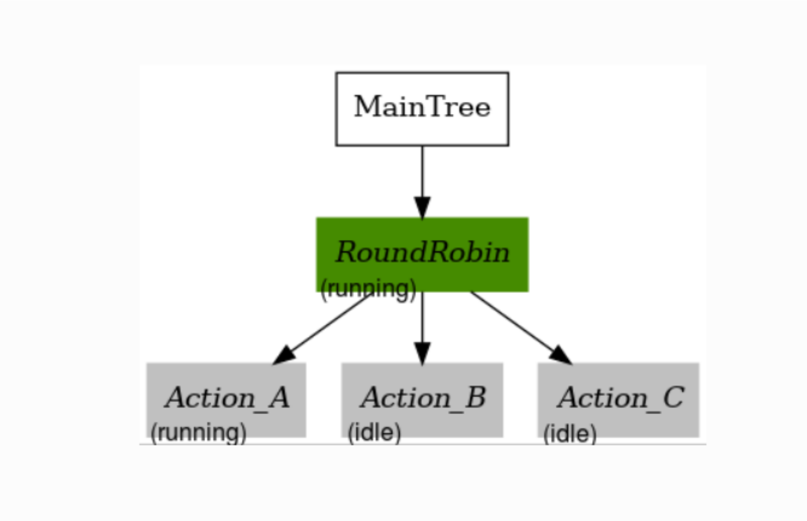
```
<root main_tree_to_execute="MainTree">
    <BehaviorTree ID="MainTree">
        <RecoveryNode number_of_retries="1">
            <ComputePathToPose/>
            <ClearLocalCostmap/>
        </RecoveryNode>
    </BehaviorTree>
</root>
```

Has **only two children** and **returns SUCCESS** if and only if the **first child** returns SUCCESS. If the **first child** returns **FAILURE**, the **second child** will be **ticked**

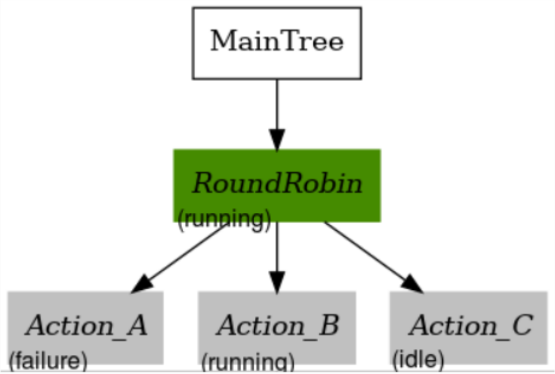
Control: RoundRobin



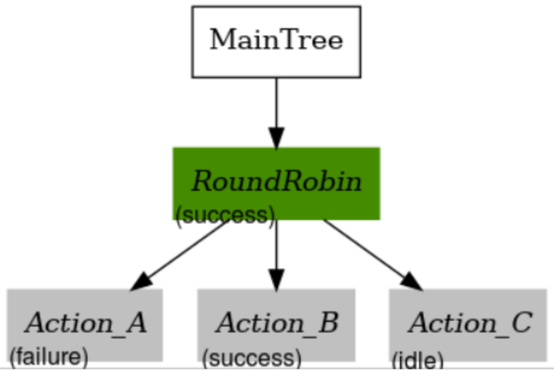
Control: RoundRobin



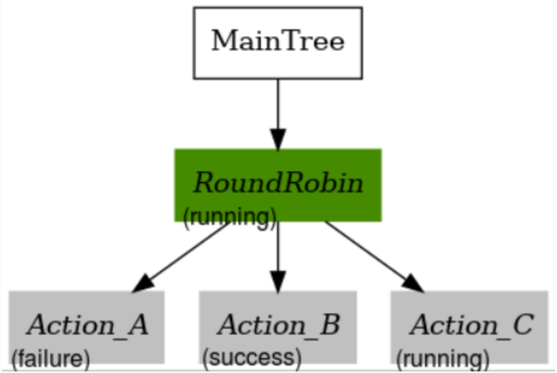
Control: RoundRobin



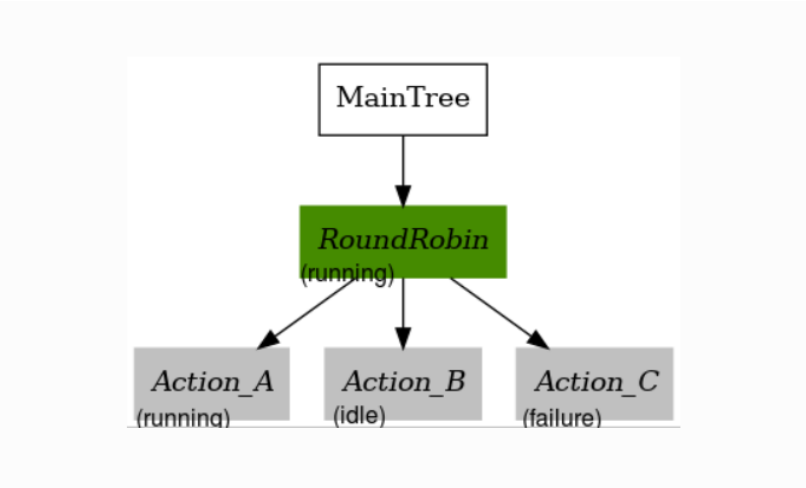
Control: RoundRobin



Control: RoundRobin

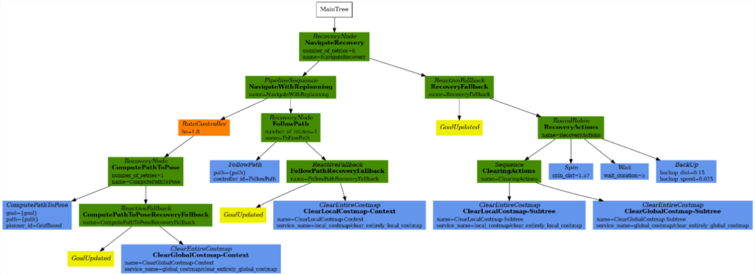


Control: RoundRobin



Navigate To Pose With Replanning and Recovery

Default BT currently used in Nav2



MainTree

RecoveryNode
NavigateRecovery
number_of_retries=6
name=NavigateRecovery

PipelineSequence
NavigateWithReplanning
name=NavigateWithReplanning

ReactiveFallback
RecoveryFallback
name=RecoveryFallback

RateController
hz=1.0

RecoveryNode
FollowPath
number_of_retries=1
name=FollowPath

GoalUpdated

RoundRobin
RecoveryActions
name=RecoveryActions

RecoveryNode
ComputePathToPose
number_of_retries=1
name=ComputePathToPose

FollowPath
path={path}
controller_id=FollowPath

ReactiveFallback
FollowPathRecoveryFallback
name=FollowPathRecoveryFallback

Sequence
ClearingActions
name=ClearingActions

Spin
spin_dist=1.57

Wait
wait_duration=5

BackUp
backup_dist=0.15
backup_speed=0.025

ComputePathToPose
goal={goal}
path={path}
planner_id=GridBased

ReactiveFallback
ComputePathToPoseRecoveryFallback
name=ComputePathToPoseRecoveryFallback

GoalUpdated

ClearEntireCostmap
ClearLocalCostmap-Context
name=ClearLocalCostmap-Context
service_name=local_costmap/clear_entirely_local_costmap

ClearEntireCostmap
ClearLocalCostmap-Subtree
name=ClearLocalCostmap-Subtree
service_name=local_costmap/clear_entirely_local_costmap

ClearEntireCostmap
ClearGlobalCostmap-Subtree
name=ClearGlobalCostmap-Subtree
service_name=global_costmap/clear_entirely_global_costmap

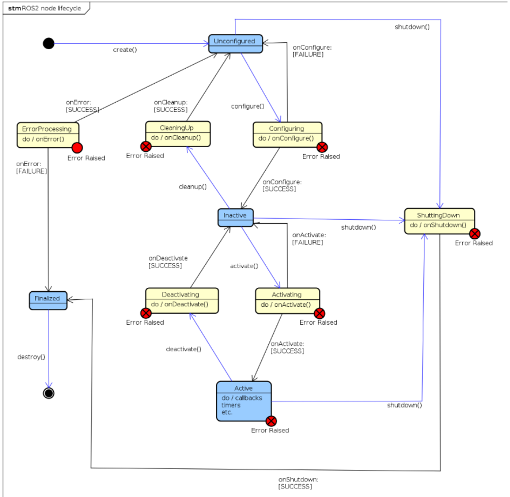
GoalUpdated

ClearEntireCostmap
ClearGlobalCostmap-Context
name=ClearGlobalCostmap-Context
service_name=global_costmap/clear_entirely_global_costmap

https://navigation.ros.org/behavior_trees/overview/detailed_behavior_tree_walkthrough.html

ROS2 Lifecycle Manager

- Allow **roslaunch** to **ensure that all components** have been **instantiated correctly** before it allows any component to begin executing its behaviour
- Also allow nodes to be restarted or replaced online
- Primary states: Unconfigured, Inactive, Active, Finalized
- Transition states: Configuring, CleaningUp, ShuttingDown, Activating, Deactivating, ErrorProcessing

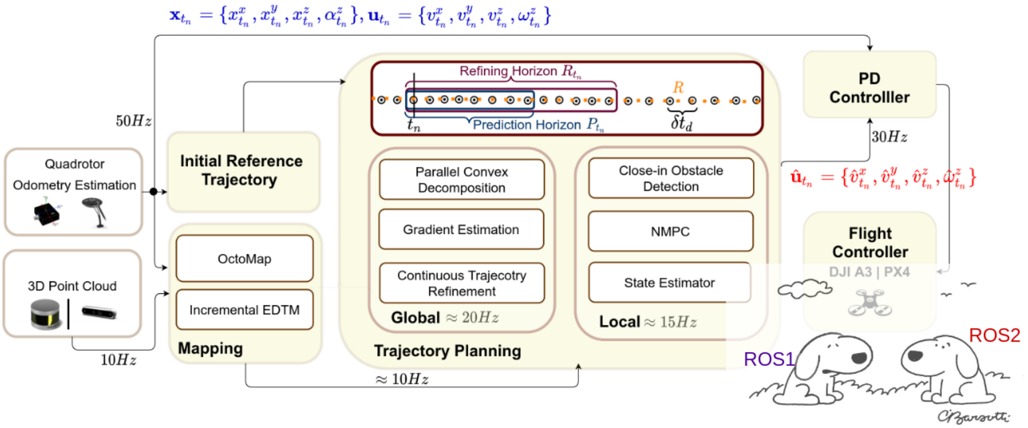


Features Status

https://docs.ros.org/en/iron/The-ROS2-Project/Features.html

As a ROS2 developer?

As a ROS2 developer?



$\mathbf{x}_{t_n} = \{x^x_{t_n}, x^y_{t_n}, x^z_{t_n}, \alpha^z_{t_n}\}, \mathbf{u}_{t_n} = \{v^x_{t_n}, v^y_{t_n}, v^z_{t_n}, \omega^z_{t_n}\}$

$50Hz$

Quadrotor Odometry Estimation

3D Point Cloud

$10Hz$

Initial Reference Trajectory

OctoMap

Incremental EDTM

Mapping

Refining Horizon R_{t_n}

R

t_n

Prediction Horizon P_{t_n}

δt_d

Parallel Convex Decomposition

Gradient Estimation

Continuous Trajecotry Refinement

Global $\approx 20Hz$

Close-in Obstacle Detection

NMPC

State Estimator

Local $\approx 15Hz$

Trajectory Planning

$\approx 10Hz$

PD Controlller

$30Hz$

$\hat{\mathbf{u}}_{t_n} = \{\hat{v}^x_{t_n}, \hat{v}^y_{t_n}, \hat{v}^z_{t_n}, \hat{\omega}^z_{t_n}\}$

Flight Controller

DJI A3 | PX4

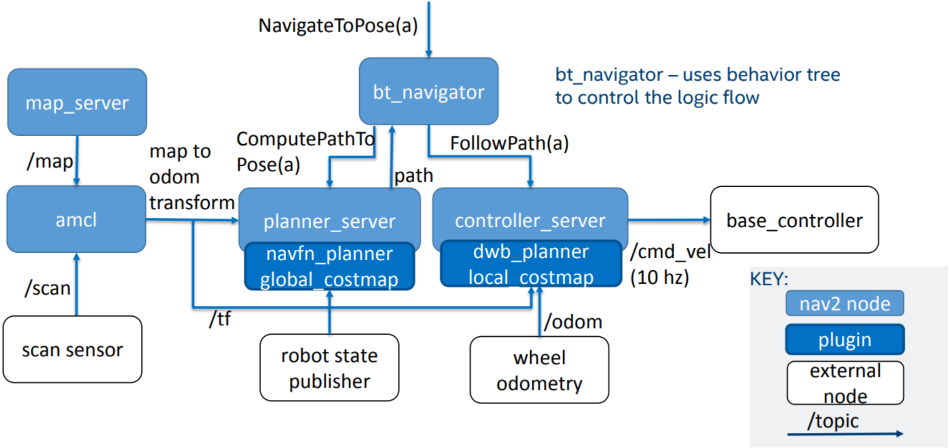
ROS1

ROS2

CBarsotti

Kulathunga, G., Hamed, H., Devitt, D., Klimchik, A. (2022). Optimization-based trajectory tracking approach for multi-rotor aerial vehicles in unknown environments. IEEE Robotics and Automation Letters, 7(2), 4598-4605.

As a ROS2 developer?



bt_navigator – uses behavior tree to control the logic flow

https://roscon.ros.org/2019/talks/roscon2019_navigation2_overview_final.pdf

CARTOONSTOCK
A CARTOON COLLECTIONS COMPANY
CBarsotti