NAMOROS: A ROS-Compatible Framework for Multi-Robot Navigation Among Movable Obstacles

David Brown[1 4], **Jacques Saraydaryan**[1 3 4], **and Olivier Simonin**[1 2 4]

1 Inria, CHROMA Team **2** INSA Lyon **3** CPE Lyon **4** CITI Laboratory

DOI: 10.xxxxxx/draft

Software

- Review
- Repository
- Archive

Editor: Open Journals

Reviewers:

- @openjournals

Submitted: 01 January 1970
Published: unpublished

Summary

NAMOROS is a ROS2 package for the problem of Navigation Among Movable Obstacles (NAMO). It enables mobile robots to plan and execute navigation tasks in environments where certain obstacles can be grasped and relocated. NAMOROS encapsulates the NAMOSIM (Renault 2020) navigation planner in a ROS2 framework to facilite the execution of NAMO plans on real and simulated robots. The package also supports environments with multiple robots and provides ROS2 services and actions to synchronize the planner state and dynamically detect and react to conflicts via NAMOSIM's communication-free coordination strategy. The package integrates seamlessly with Gazebo Sim and any ROS2-compatible mobile robot platform such as Turtlebot, and supports holonomic and differential-drive motion models. NAMOROS is designed for research and development in multi-robot navigation, path planning, and socially-aware navigation, particularly in interactive environments.

Statement of Need

Robotic navigation in dynamic and cluttered environments remains a critical challenge in robotics. Traditional navigation methods typically assume static obstacles, but many real-world applications require robots to actively interact with and manipulate obstacles to achieve their objectives. NAMOROS takes a significant first step toward addressing this challenge by providing open-source tools for Navigation Among Movable Obstacles (NAMO), thus enabling reproducible research in and practical deployment of NAMO algorithms on real robotic systems.

In addition to single-robot navigation, NAMOROS provides important multi-robot conflict avoidance and deadlock resolution capabilities. It continuously updates the planner with the current state of the environment, detects conflicts between robots, and makes communication-free decisions to resolve them. This makes it particularly suitable for multi-robot systems operating in shared, dynamic spaces, where reactive and adaptive behavior is essential for robust operation.

Software Description

NAMOROS consists of:

- ROS2 nodes for computing and executing NAMO plans
- Integration with the `namosim` [1,2] planner for simulation
- Support for holonomic and differential drive robots
- Extensible agent framework (e.g., Stilman2005 baseline agent)
- Tools for visualization (RViz), scenario creation, and benchmarking
- Example scripts and demonstration scenarios

Key Features

NAMOROS provides the following key features:

- **Full ROS2 Compatibility**: Includes modular ROS2 nodes for NAMO planning and execution.
- **Multi-Robot Support**: Supports communication-free multi-robot coordination by dynamically detecting and reacting to conflicts.
- **Movable Obstacle Interaction**: Enables grasping, relocating, and releasing of obstacles during navigation.
- **Dynamic Replanning**: Continuously updates plans based on real-time sensor data and robot interactions.
- **Simulation Integration**: Seamlessly integrates with Gazebo Sim for physics-based validation.
- **Support for Multiple Locomotion Models**: Compatible with both holonomic and differential-drive robots.
- **Behavior Tree Architecture**: Modular, extensible control logic implemented with behavior trees.
- **Custom Planning Backend**: Interfaces with `namosim` [1,2], a dedicated NAMO planner with multi-robot awareness.
- **Extensive Visualization Tools**: Includes support for RViz and custom visualization markers.
- **Scenario and Benchmarking Tools**: Offers utilities for reproducible experiments and comparative evaluation.
- **Custom Gazebo Plugins**: Includes plugins to simulate physical interactions like grabbing and releasing obstacles.

Architecture

The system is organized as ROS2 packages:

- `namoros`: ROS2 nodes to control the robot and interacting with the namosim [1,2] planner within a behavior tree framework
- `namosim`: The core planner for navigation and multi-robot coordination
- `namoros_msgs`: Custom ROS2 message definitions
- `namoros_gz`: Custom Gazebo plugin for simulating grab and release actions.

Here is a block diagram showing the main components of NAMOROS:

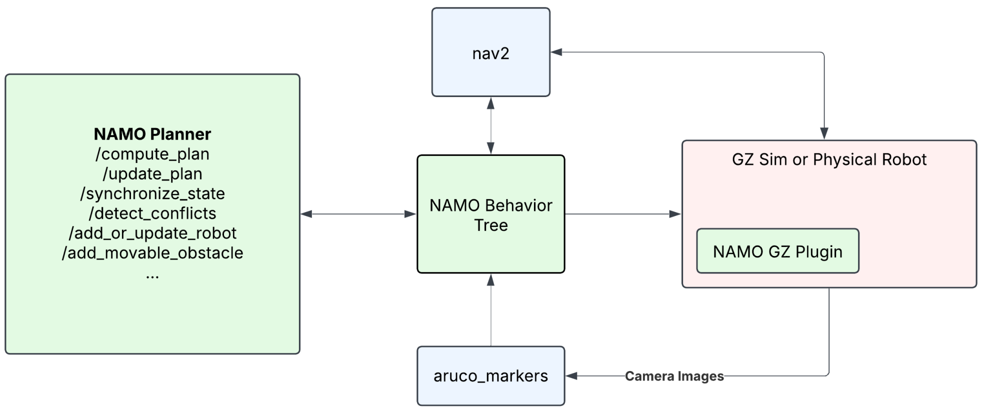

Figure 1: NAMOROS Architecture

The **NAMO Planner** block is a custom ROS2 node that manages the namosim [1,2] planner and exposes services and actions for interacting with it.

The **NAMO Behavior Tree** block is another custom node that executes the main behavior tree which controls the robot execution and interaction with the planner node.

The other blocks, `nav2` and `aruco_markers`, are third-party packages used for simple navigation and detection of visual markers placed on movable obstacles.

If running in a Gazebo simulation, a plugin from the `namoros_gz` package is provided to simulate grab and release actions. It works by dynamically creating a fixed joint between a user-chosen link on the robot and a link on the obstacle.

Main Behavior Tree

NAMOROS provides a behavior tree to run NAMO plans on a real or simulated robot. This behavior tree controls the robot while managing sensor data and interactions with the planner node. The tree is illustrated in the following diagram and is configured to tick at a frequency of 2Hz. The robot starts by waiting to receive a start pose and a goal pose. These may come from the scenario file or be published to the corresponding ROS2 topics.

The behavior tree continuously monitors the robot's sensor data to track the positions of other robots and movable obstacles. It uses this data during specific periods to synchronize the planner node's state with the estimated state of the environment, which is necessary for conflict detection. The *New Movable* node encapsulates a subtree that handles dynamic detection of movable obstacles but is only used when that feature is activated and is not illustrated for brevity.

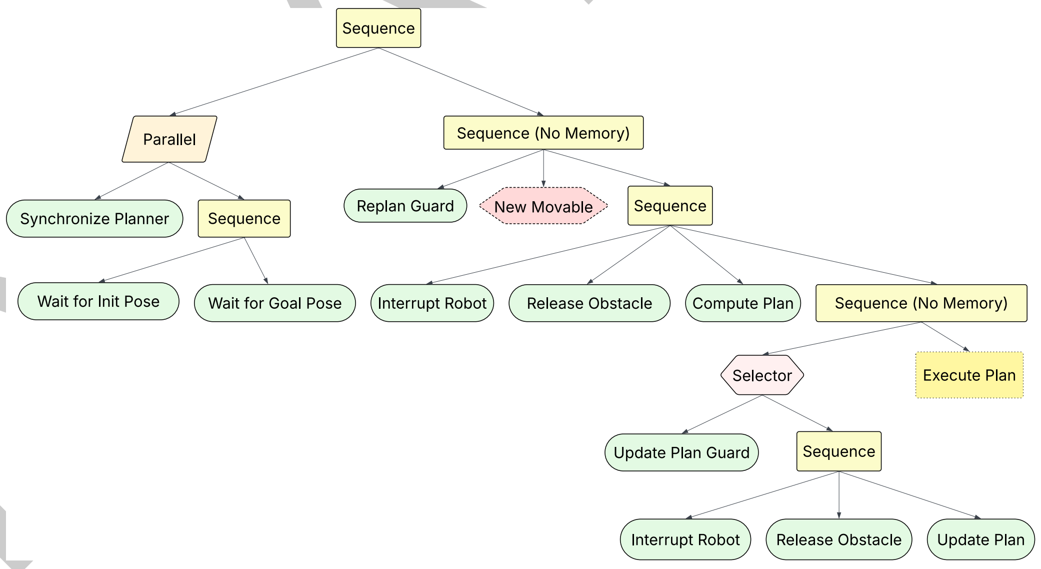

Figure 2: Main Behavior Tree

Execute Plan Subtree

Because a NAMO plan consists of multiple behaviors such as path following, and grabbing and releasing obstacles, and because the plan is initially unknown and subject to change, the *Execute Plan* behavior dynamically creates and executes a subtree corresponding to the current plan. The following diagram shows an example subtree which consists of a *transit* path followed by a *transfer* path to move an obstacle, and lastly another transit path to reach the goal. Immediately before and after each *transfer* path there are also grab and release sequences. Each of these behaviors are themselves small subtrees. The *Execute Plan* subtree always starts with a release behavior just in case the robot was already holding an obstacle at the time the plan was computed.

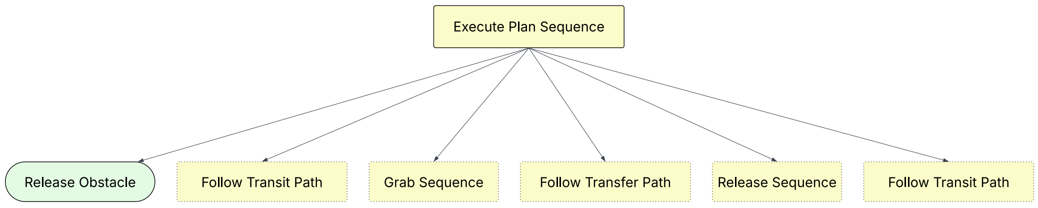

Figure 3: Execute Plan Tree

Conflict Avoidance and Deadlock Resolution

Detecting and handling conflicts relies on implicit-coordination strategy implemented in NAMOSIM's `Stilman2005` agent. However, it depends on keeping the planner synchronized with the robot's current perceived state. During path following, the behavior tree periodically synchronizes the planner node with the current estimated state of the environment and checks for conflicts. When a conflict is detected, the robot is interrupted, the planner node is manually synchronized with the robot's current perceived state, the plan is *updated*, and finally plan execution is restarted. The deadlock resolution strategies likewise depend on the coordination algorithm from the `Stilman2005` agent in NAMOSIM.

Acknowledgements

This research was supported by an Inria ADT initiative. We express our gratitude to Benoit Renault, whose PhD thesis forms the foundation of this work.

License

References

Renault, B., Saraydaryan, J., Brown, D., & Simonin, O. (2024). Multi-Robot Navigation among Movable Obstacles: Implicit Coordination to Deal with Conflicts and Deadlocks. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. https://hal.science/hal-04705395

Renault, B., Saraydaryan, J., & Simonin, O. (2020). Modeling a Social Placement Cost to Extend Navigation Among Movable Obstacles (NAMO) Algorithms. *IEEE/RSJ IROS 2020*. https://doi.org/10.1109/IROS45743.2020.9340892