Cleaning Toys

Room organization using mobile manipulation

Anshul Jindal

Tanish Mishra

Evan Arenburg

PROBLEM

Scattered toys pose safety risks, leading to parental concerns, and a loss of quality time.

1500 times/year

parents pick up after kids.

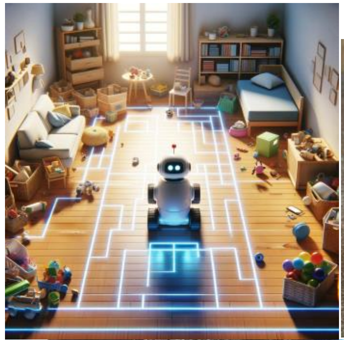
4/5 parents say

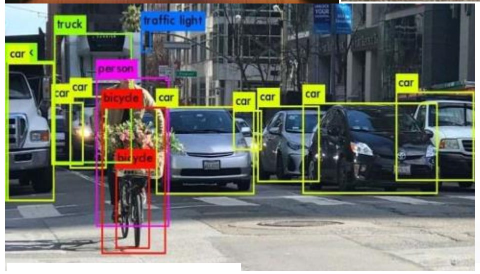
kids misplace things they put away.

71% of parents injured

from stepping on toys.

https://swnsdigital.com/us/2017/06/parents-have-to-pick-up-after-their-kids-1500-times-a-year/





APPROACH

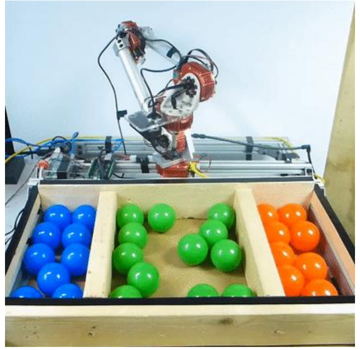
Hardware

- Mobile robot
  - Differential Drive
  - LIDAR
  - Intel RealSense Stereo Camera
- Robot Arm

Software

- Object Detection
- Motion Planning
- Grasp Planning

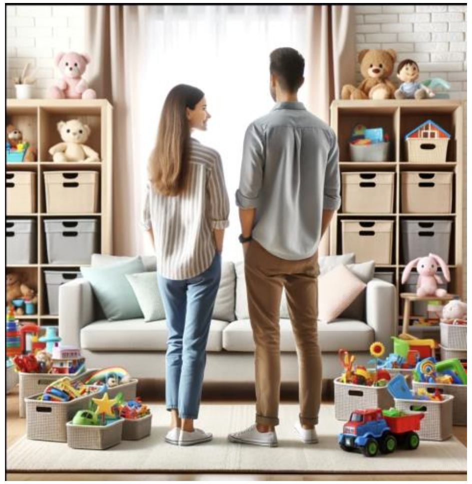
ROS 2 NVIDIA OMNIVERSE MoveIt2



RESULT

Individually simulate different aspects of a robot that can autonomously detect and sort objects:

1. **Object detection:** to identify target items and obstacles in a cluttered environment
2. **Robotic arm planning:** to simulate object pick-and-place
3. **Differential drive robot:** that can successfully navigate around a cluttered room between identified objects and target bins



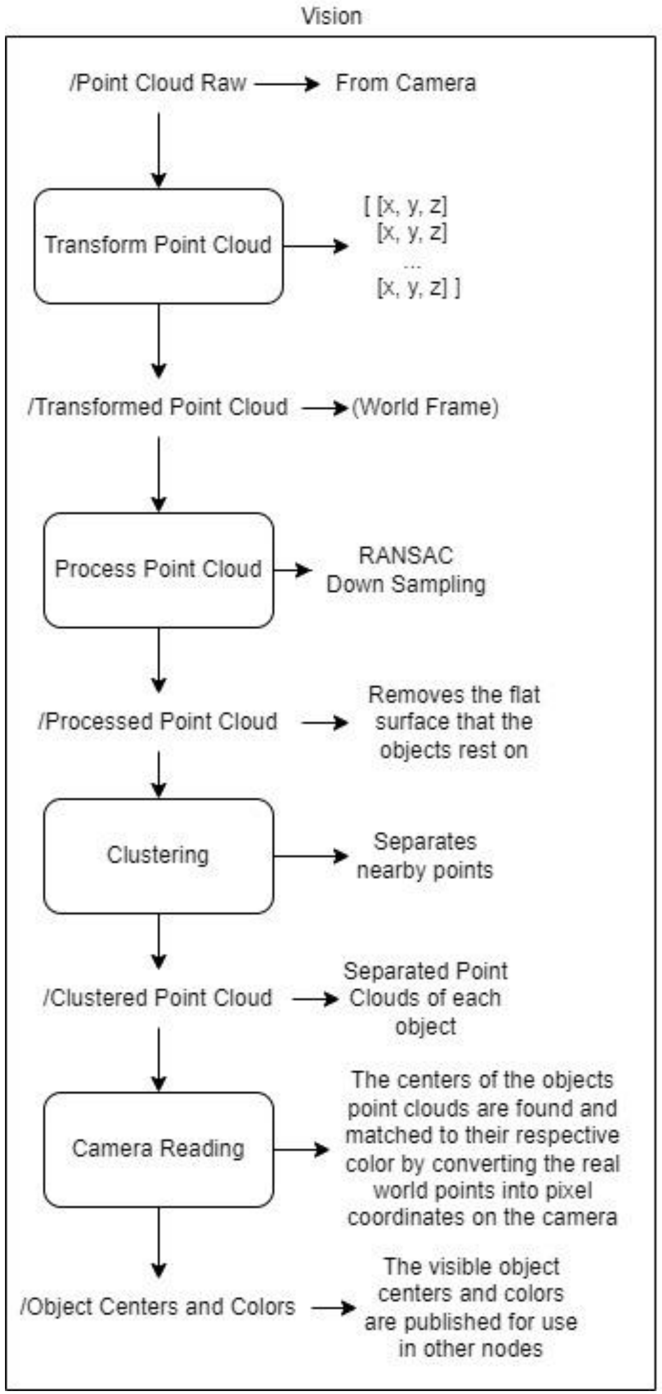
IMPACT

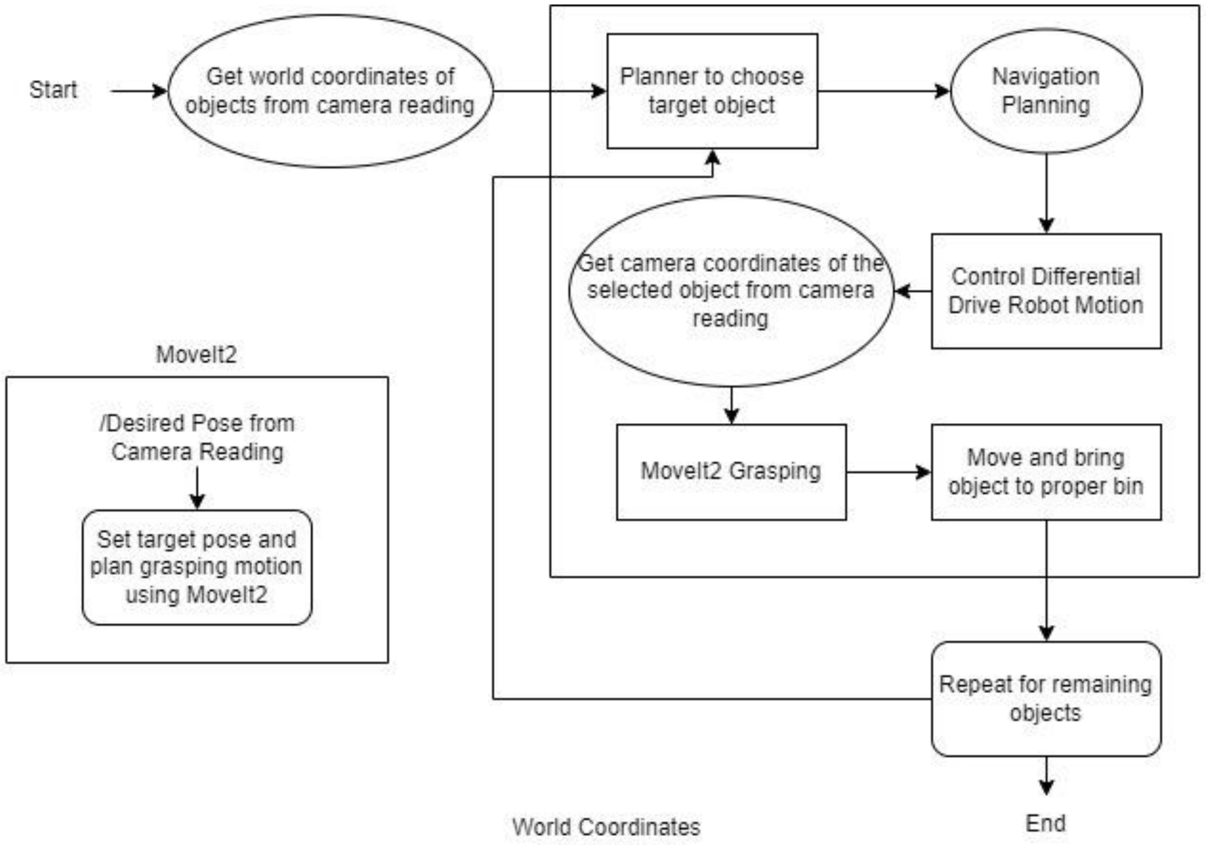
Reduced burden of cleaning up, decreased mental stress. Automating menial cleaning tasks frees up time for family.

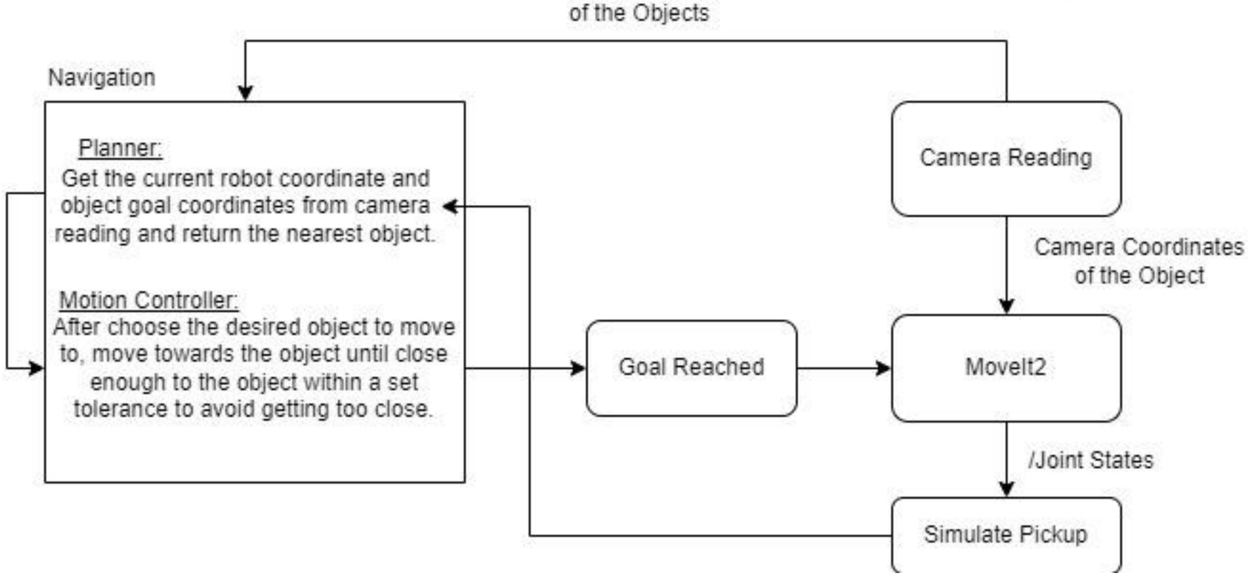
Project idea can be extended to:

- Assistive robotics
- Agriculture automation
- Emergency & Disaster Response

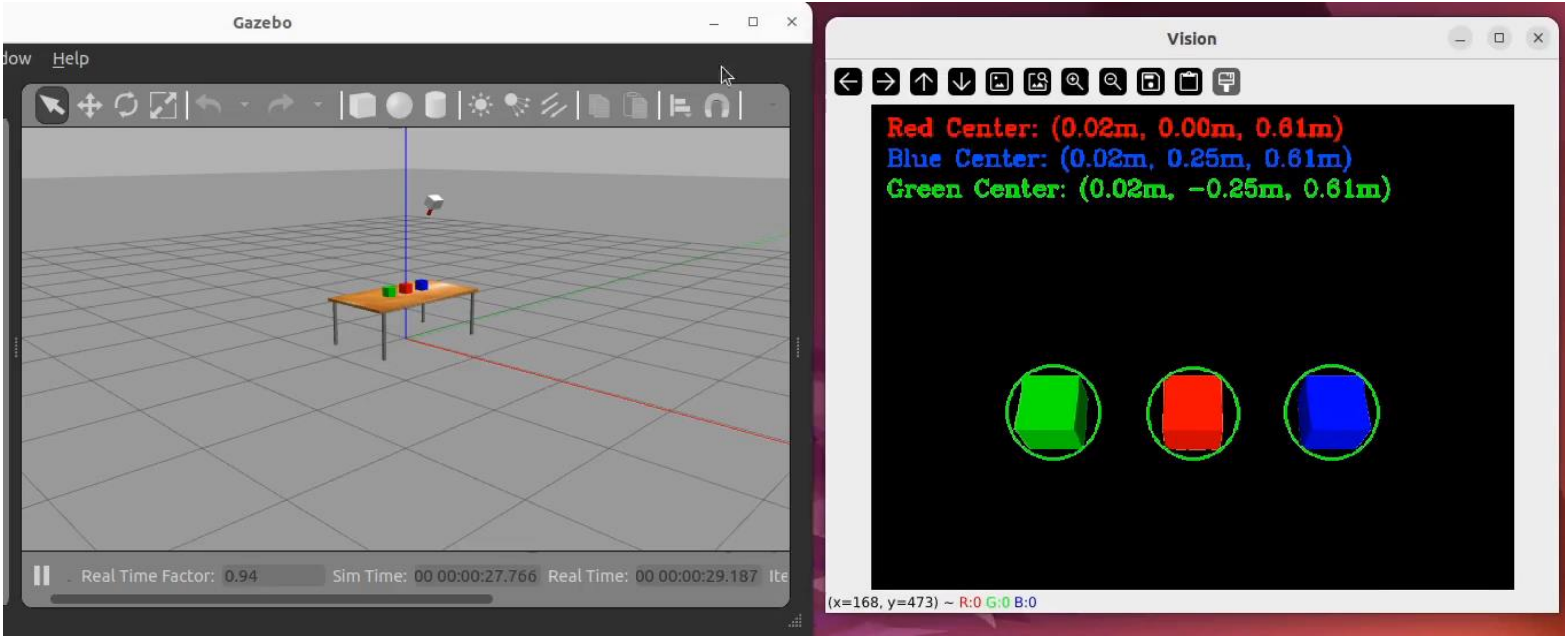
Design Flow Chart:







Vision:

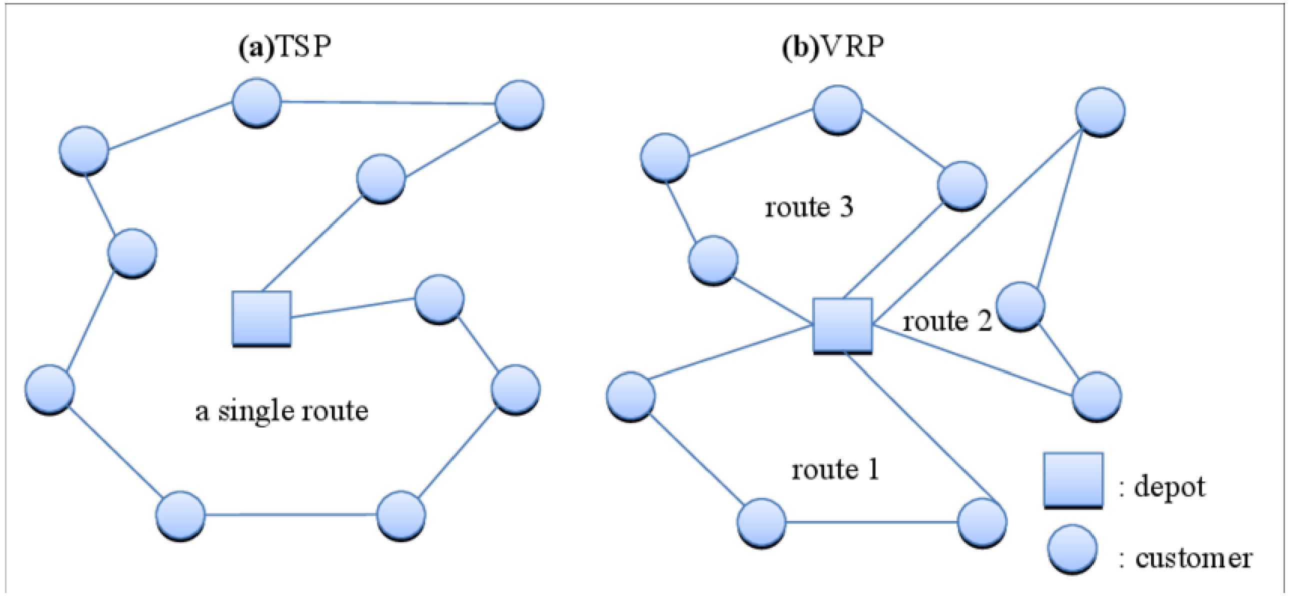

Gazebo
dow Help
Real Time Factor: 0.94 Sim Time: 00 00:00:27.766 Real Time: 00 00:00:29.187 Ite
Vision
Red Center: (0.02m, 0.00m, 0.61m)
Blue Center: (0.02m, 0.25m, 0.61m)
Green Center: (0.02m, -0.25m, 0.61m)
(x=168, y=473) ~ R:0 G:0 B:0

Vision:

- Detects the centers of the red, blue, and/or green object(s) and finds the pixel center of the object(s).
- Uses the point cloud from the camera by first transforming the points into the world frame from the camera frame.
- The transformed point cloud is processed to remove the flat plane that the object(s) rest on.
- The processed point cloud is clustered into separate objects by collecting nearby points.
- The object center(s) of each cluster are found and converted into pixel coordinates as seen by the camera.
- The pixel centers as found in the first part are compared to the pixel coordinates of the cluster centers and then these pixel coordinates are matched to each other.
- The centers of each object are found with an error of about ± 0.03m and are associated with their respective detected color.

Background

- Vehicle Routing Problem (An extension of the Travelling Salesman Problem)
    - Involves determining optimal routes for multiple vehicles to service a set of locations,
- Traveling Salesman Problem focuses on finding the shortest route to visit all locations exactly once and return to the starting point.
- VRPPD is the VRP with pickup and delivery



Minimizing the Carbon Footprint for the Time-Dependent Heterogeneous-Fleet Vehicle Routing Problem with Alternative Paths - Scientific Figure on ResearchGate. Available from: https://www.researchgate.net/figure/Illustration-of-the-traveling-salesman-problem-TSP-and-vehicle-route-problem-VRP_fig1_277673931

Background

- Main Takeaways
  - Scope for novelty in application of task parallelization for path search
  - Energy usage minimization has not been very well researched
  - Not much VRP research in the context of household robotics
  - Dynamic VRP is also a very active area of research

NVidia Isaac Sim vs ROS Gazebo

Simulation Fidelity:

Isaac Sim:

-high-fidelity physics

-optimized for Nvidia GPU

-realistic simulations

-needs resources required utilizes Nvidia PhysX engine for simulations

-well-suited for real-time applications

Gazebo:

-well-integrated with ROS

- multiple physics engines offered, like ODE, Bullet, Simbody, and DART

User Interface and Usability:

Isaac Sim:

-Modern, user-friendly interface

-support for visual scripting

-faster prototyping without deep programming knowledge.

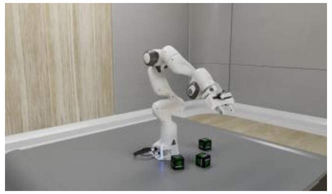
-bridges the gap between simulation and real-world deployment

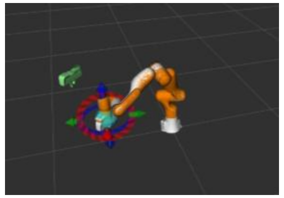
-less support in community on usage. Documentation is sparse

Gazebo:

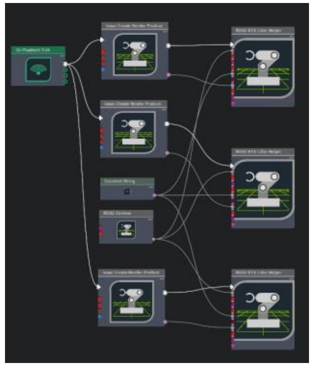
-traditional scripting and manual configuration

-higher control over simulation configs

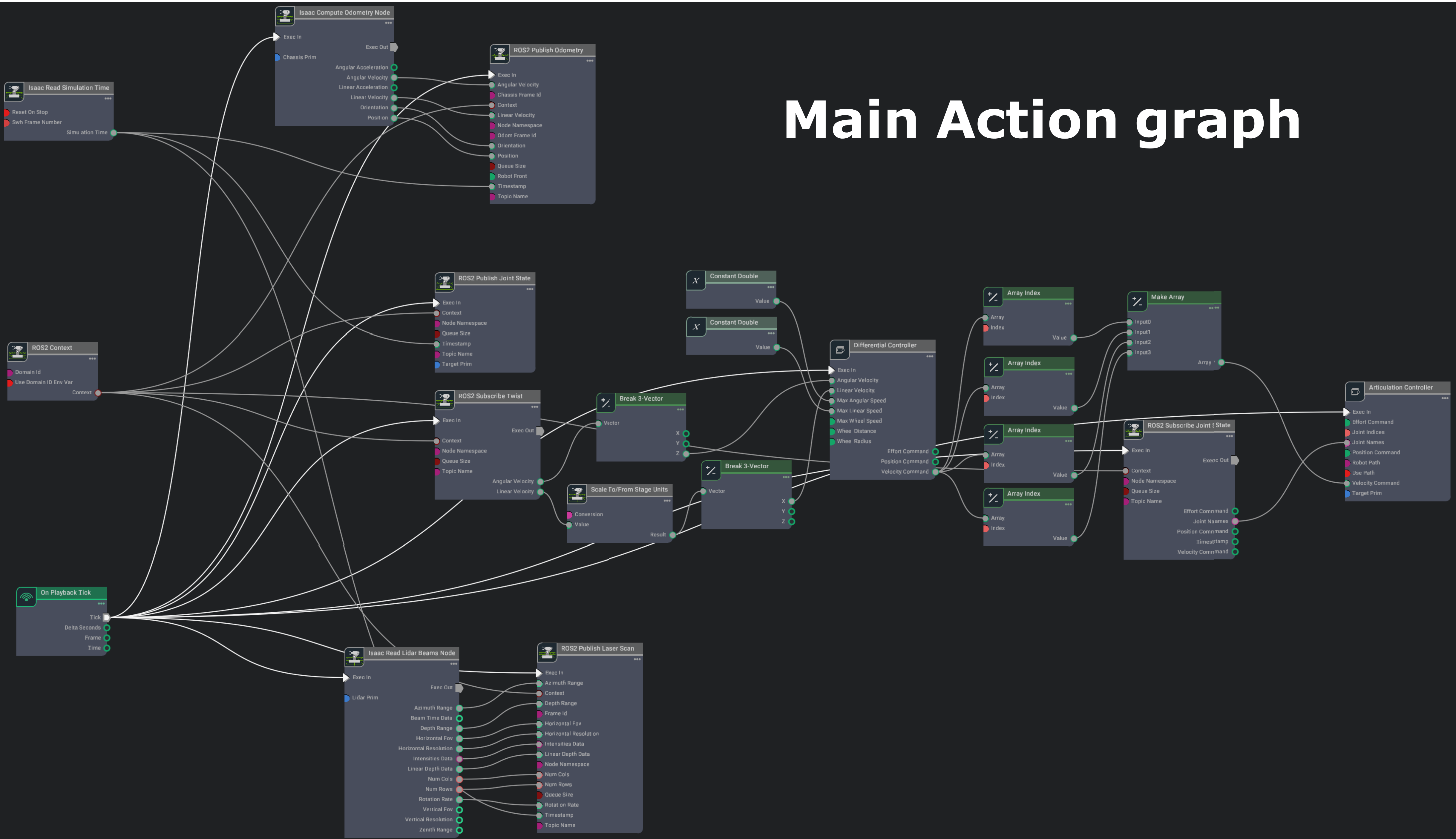




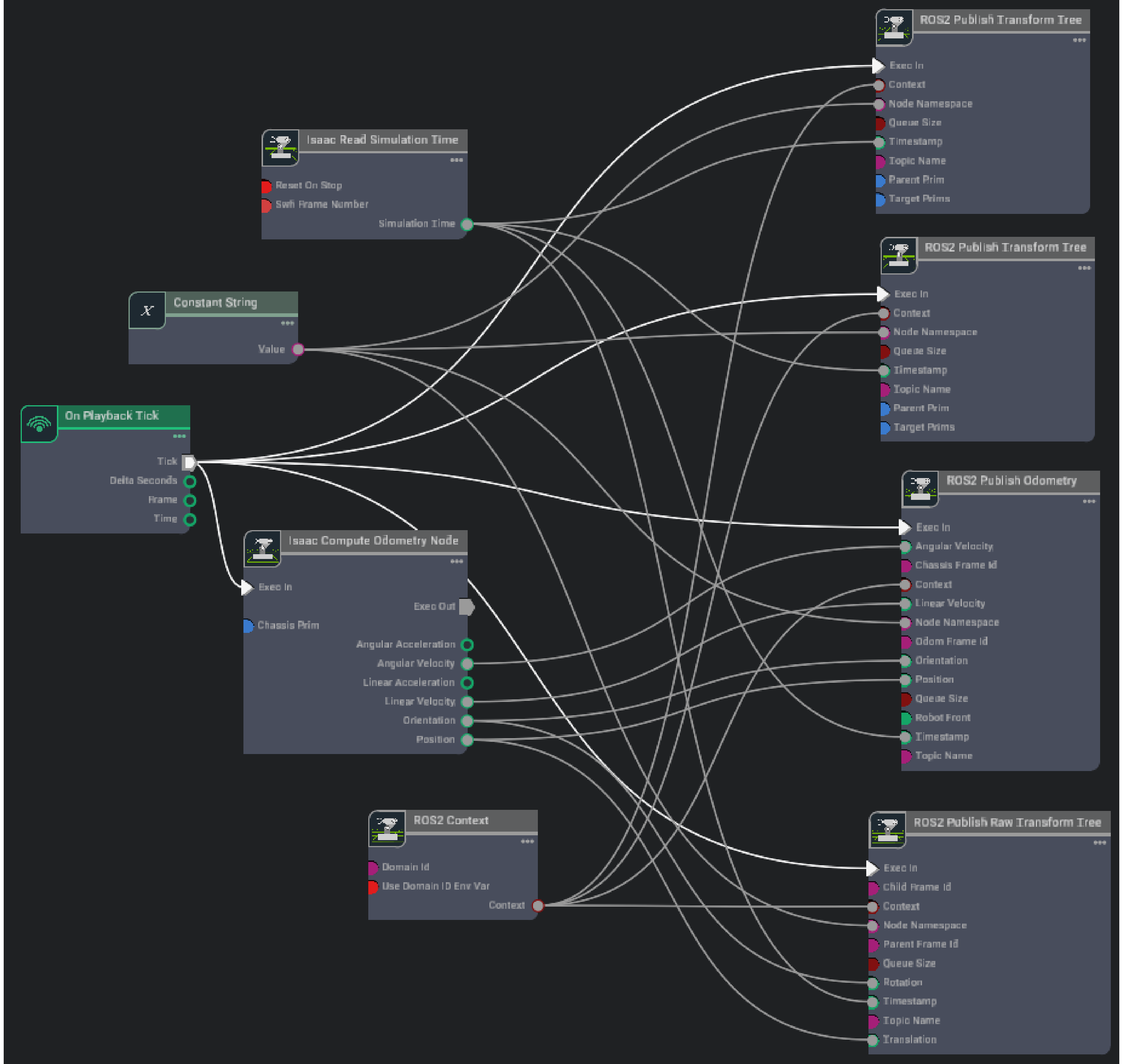
```xml
<?xml version="1.0" ?>
<robot name="ur5e" xmlns:xacro="http://ros.org/wiki/xacro">

  <link name="base_link">
    <visual>
      <geometry>
        <mesh filename="package://robot/meshes/ur5/visual/base.dae"/>
      </geometry>
      <material name="LightGrey">
        <color rgba="0.7 0.7 0.7 1.0"/>
      </material>
    </visual>
    <collision>
      <geometry>
        <mesh filename="package://robot/meshes/ur5/visual/base.dae"/>
      </geometry>
    </collision>
    <inertial>
      <mass value="4.0"/>
      <origin rpy="0 0 0" xyz="0.0 0.0 0.0"/>
      <inertia ixx="0.00443333156" ixy="0.0" ixz="0.0" iyy="0.00443333156" iyz="0.0" izz="0.0072"/>
    </inertial>
  </link>


  <joint name="shoulder_pan_joint" type="revolute">
    <parent link="base_link"/>
    <child link="shoulder_link"/>
    <origin rpy="0.0 0.0 0.0" xyz="0.0 0.0 0.163"/>
    <axis xyz="0 0 1"/>
    <limit effort="150.0" lower="-6.283185307179586" upper="6.283185307179586" velocity="3.14"/>
    <dynamics damping="0.0" friction="0.0"/>
  </joint>
```



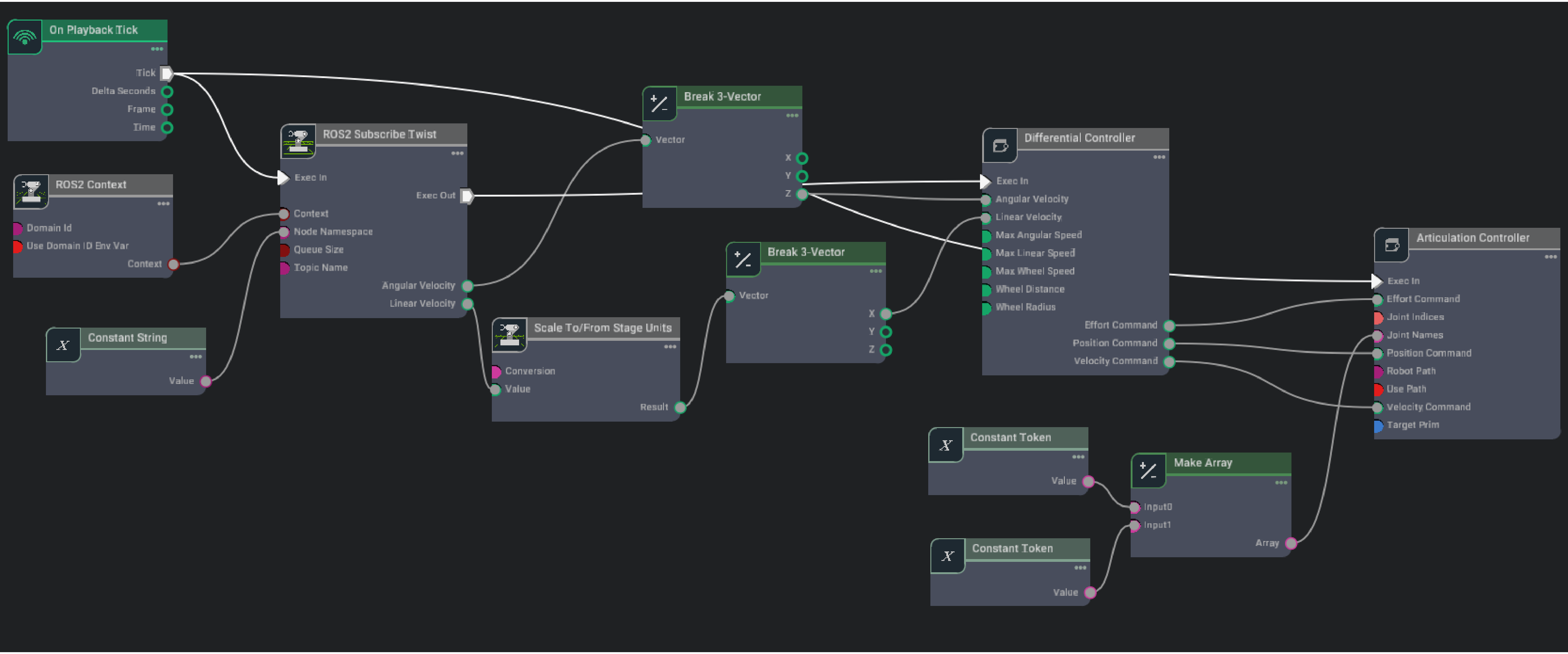
Main Action graph



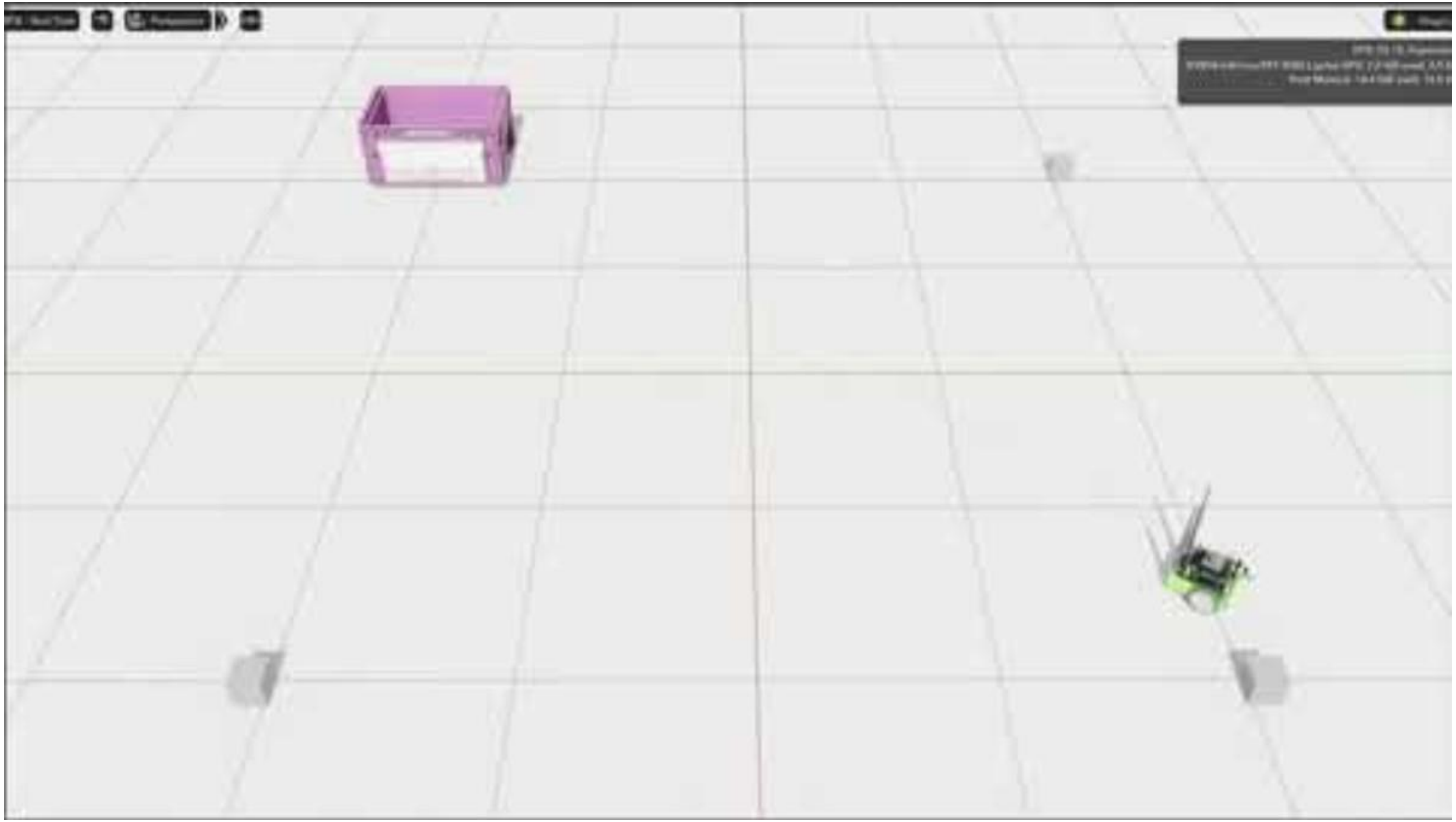
Action graph: Transform Tree and Odometry



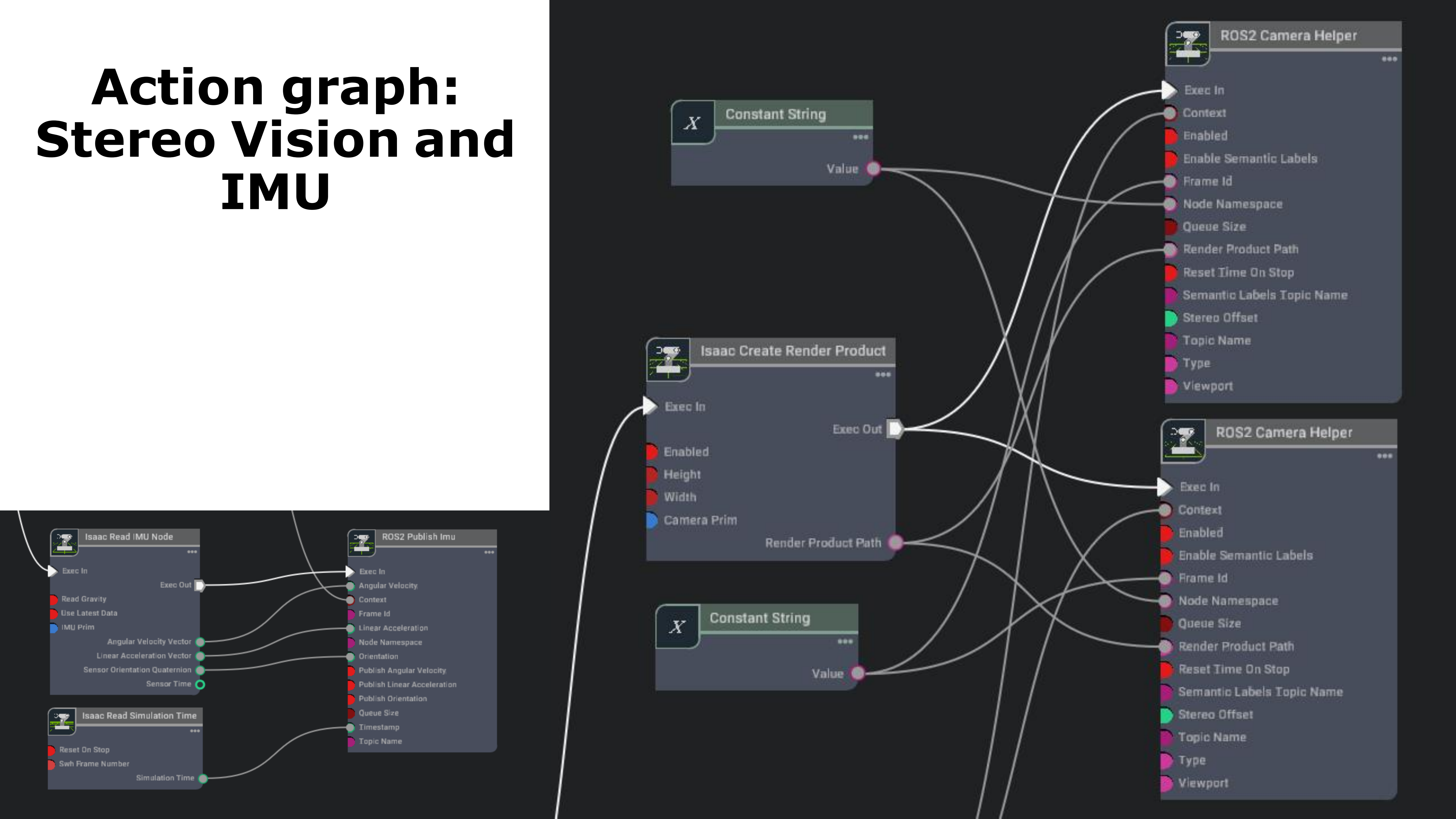
Action graph: Differential Drive



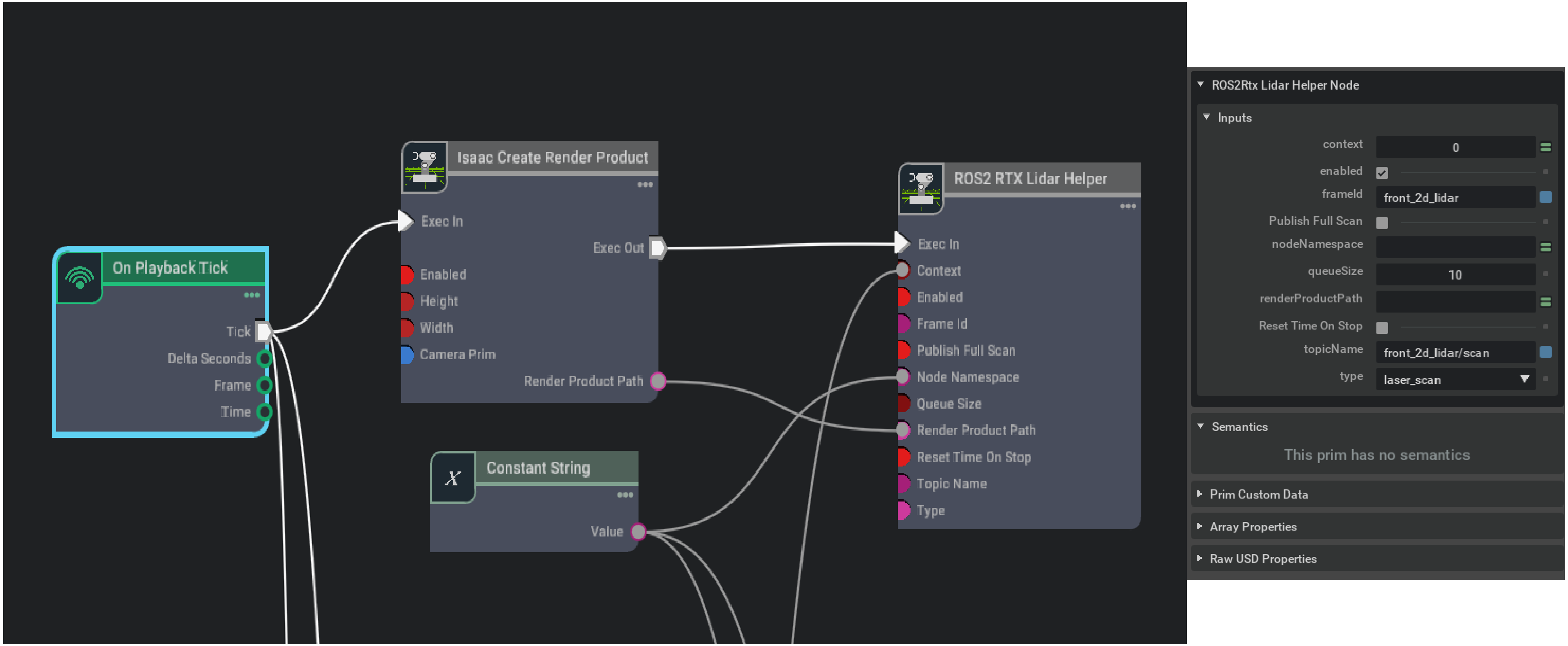
Waypoint Navigation



Action graph: Stereo Vision and IMU



Action graph: ROS RTX Lidars



Thank you!