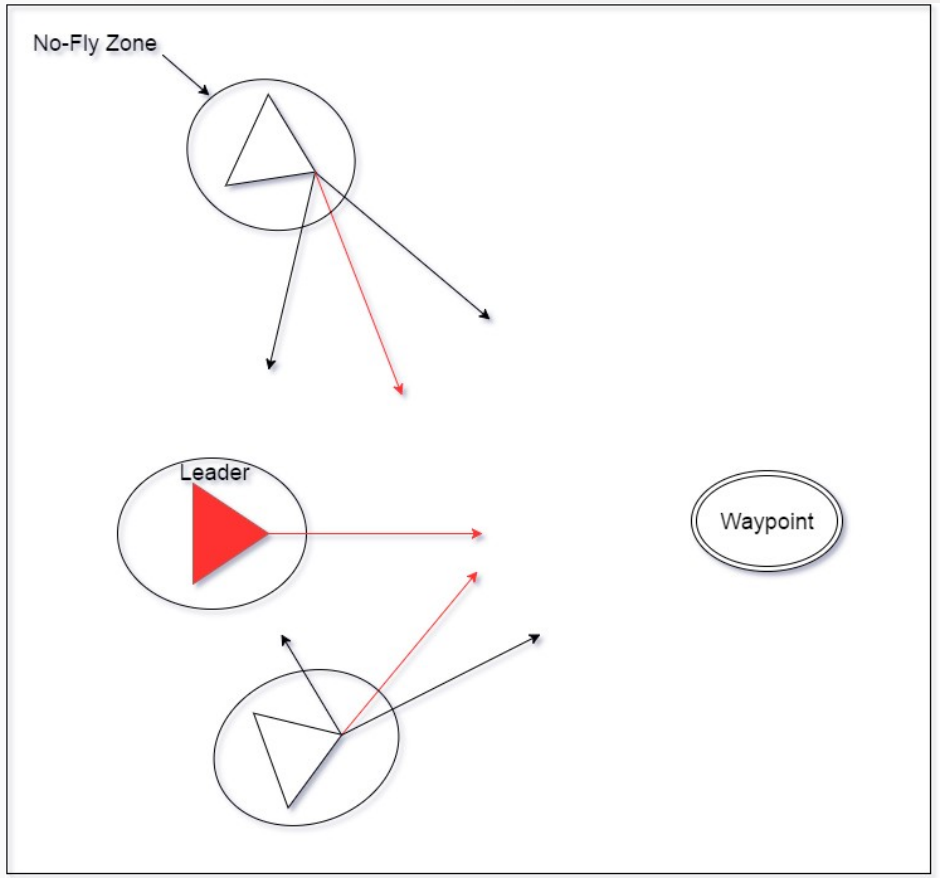
CS491/691: Introduction to Aerial Robotics

Simulation-based Control of Multirotor Aerial Vehicles

Amirhesam Yazdi, Andrew Menard, Tyler Goffinet, Vladislav Savranschi

Motivation and Problem Description

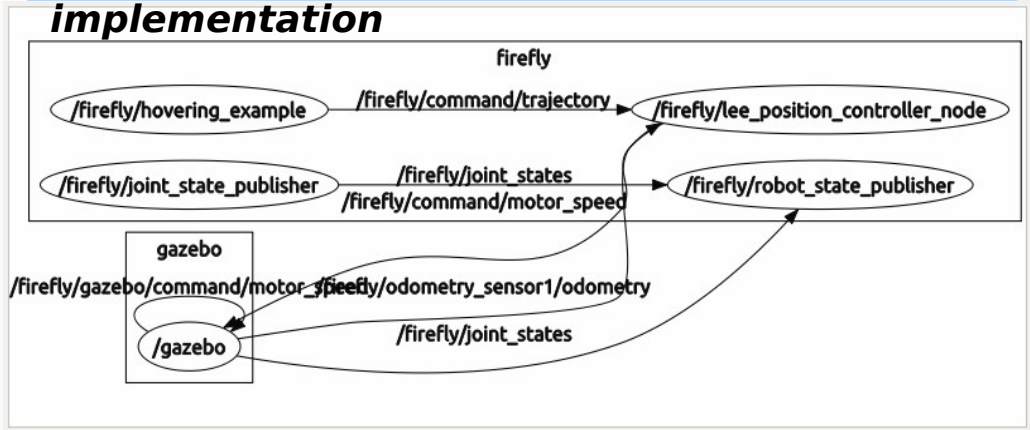
- Original Idea: Drone Bumper Cars
- Evolved into swarm collision avoidance
- Virtual, dynamically sized bubbles that prevent collisions
- Drones follow a “leader” to a waypoint
- Math is fairly easy
  - Get a vector from drone to waypoint
  - Add that vector to a vector to the leader
  - If drone positions will be closer than threshold on next tick, set short waypoint away, then return
- Need to figure out ROS, and RotorS first

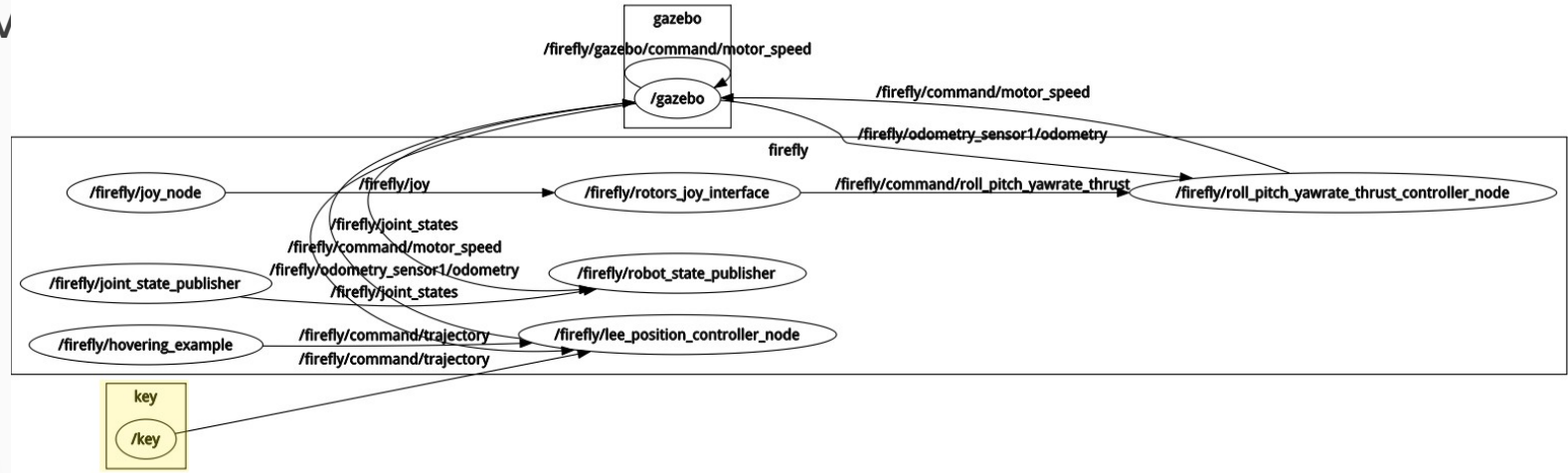


Proposed Approach

- Understanding Robot Operating System (ROS) and RotorS simulator
- Interacting with the waypoint publisher to implement easy go-to flight controls
- Creating a ROS package to publish waypoint commands to the MAV
- Reverse engineering the existing joystick interface code to implement keyboard controls for the MAV

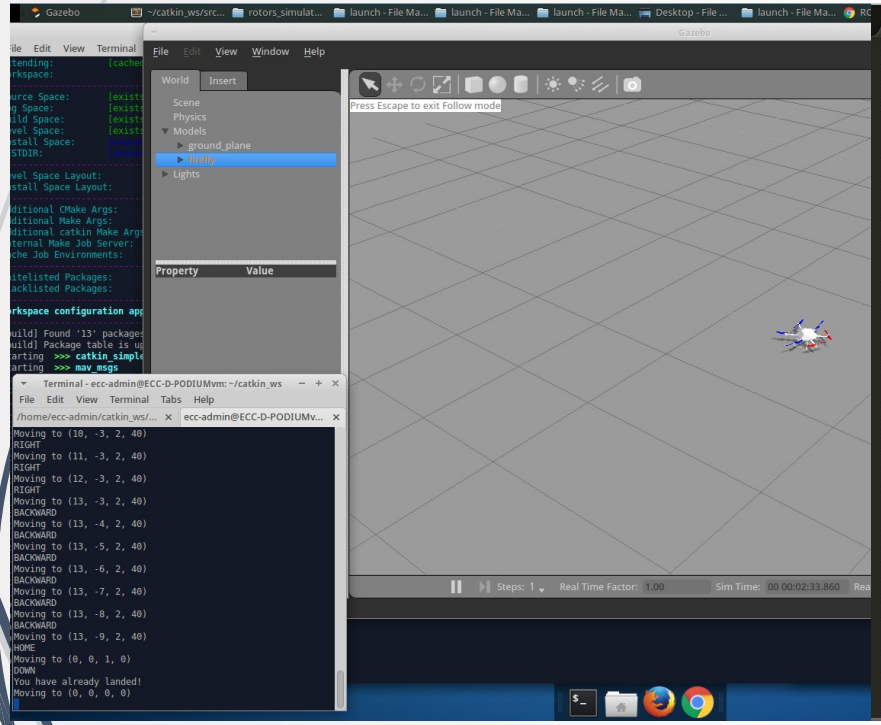
Rqt-Graph: Old (Top) vs. New (Bottom) implementation

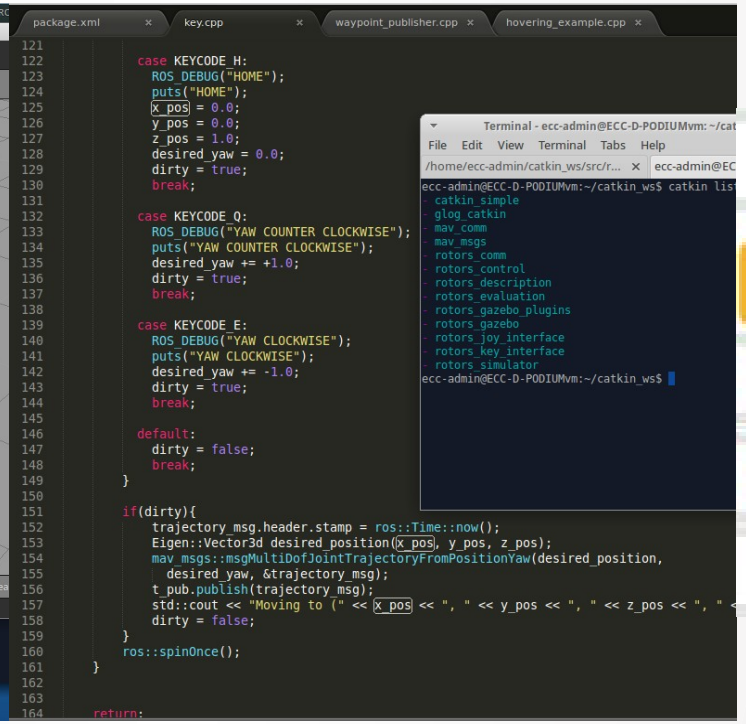


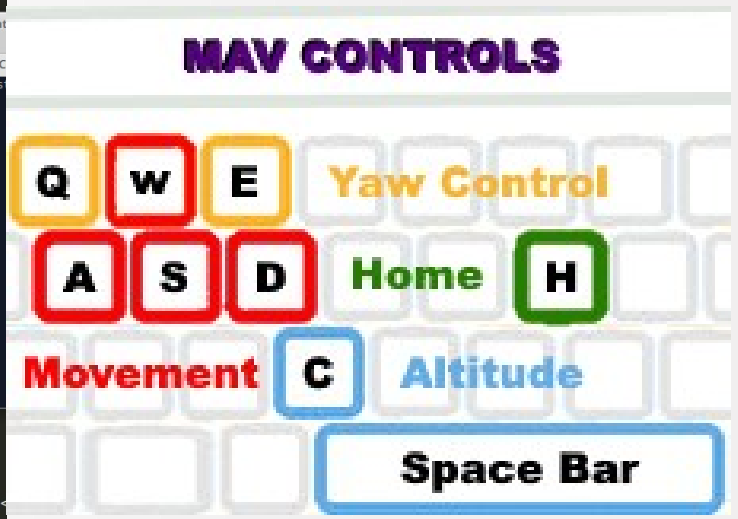


System Description

- Robot Operating System - provides libraries and tools to help software developers create robot applications.
- RotorS - MAV gazebo simulator that runs on ROS. Has the ability to simulate many MAVs in many scenarios.







Results

- Comprehensive ROS/RotorS package and node
  - ‘rotors_key_interface’
- User-friendly keyboard controls
- Full control of MAV axes (θ,φ,ψ)
- Directs the MAV to positions relative to the current location