Introduction to Gazebo

Evan Krell

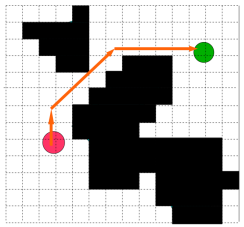
Texas A&M University - Corpus Christi

June 2019

Outline

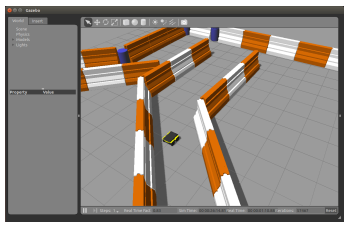
Problem Variations

Experiments: Real or Simulated



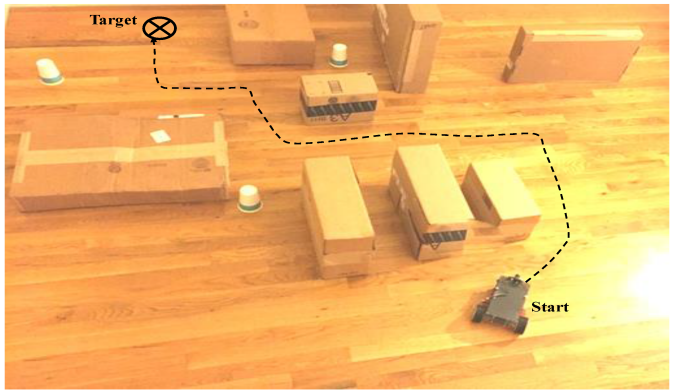
Simulated: 2D

- Simple 2D kinematics.
- Applicable to numerous robots.
- solutions may be unrealistic.



Simulated: 3D

- Closer to real life, model dynamics.



Real

- Most realistic.
- Very time-consuming.
- May require engineering skills.
- Forced to address real-world limitations.
- Most impressive when completed.

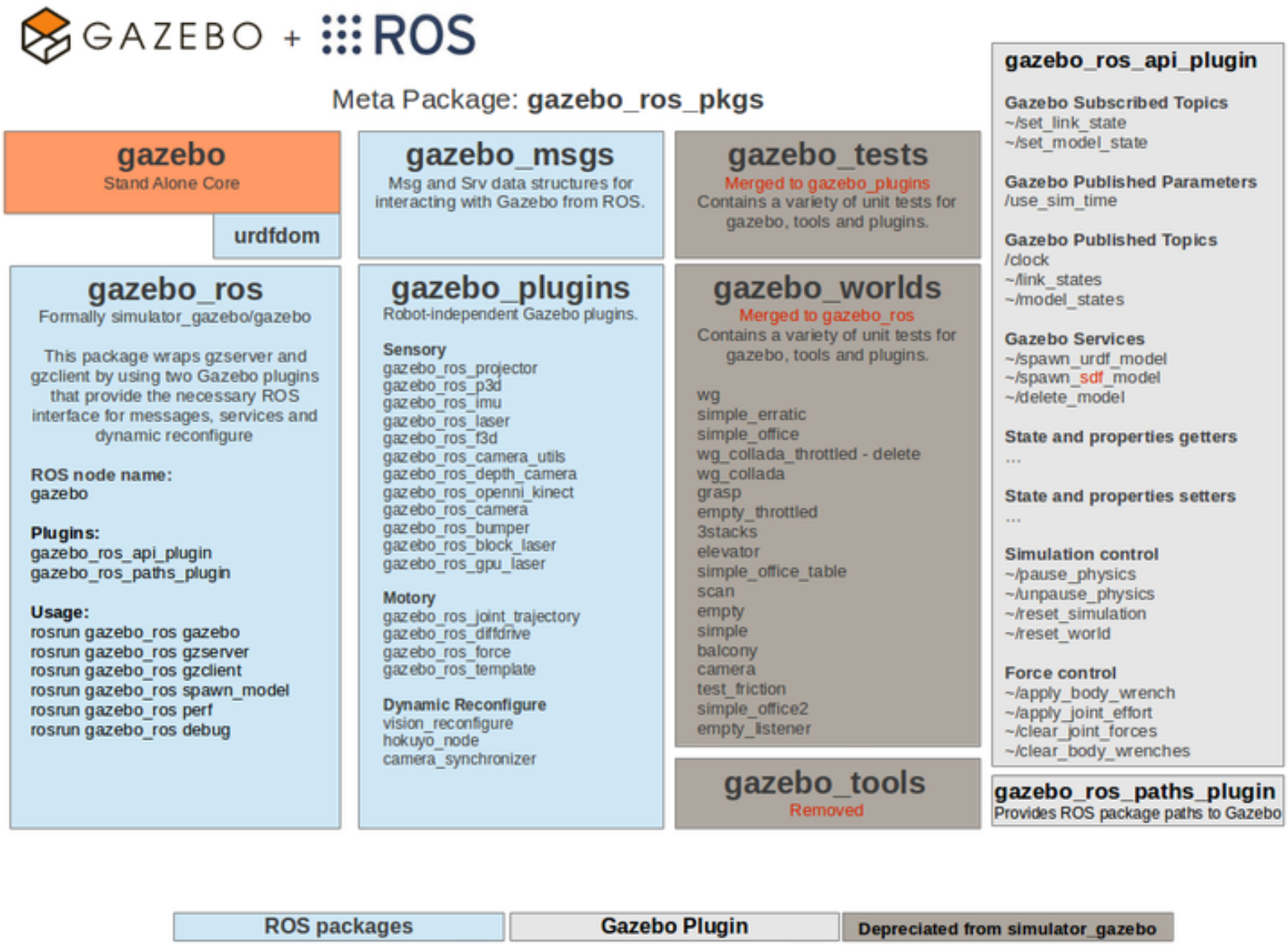
Gazebo

Gazebo is 3D physics-based simulator for robotics.

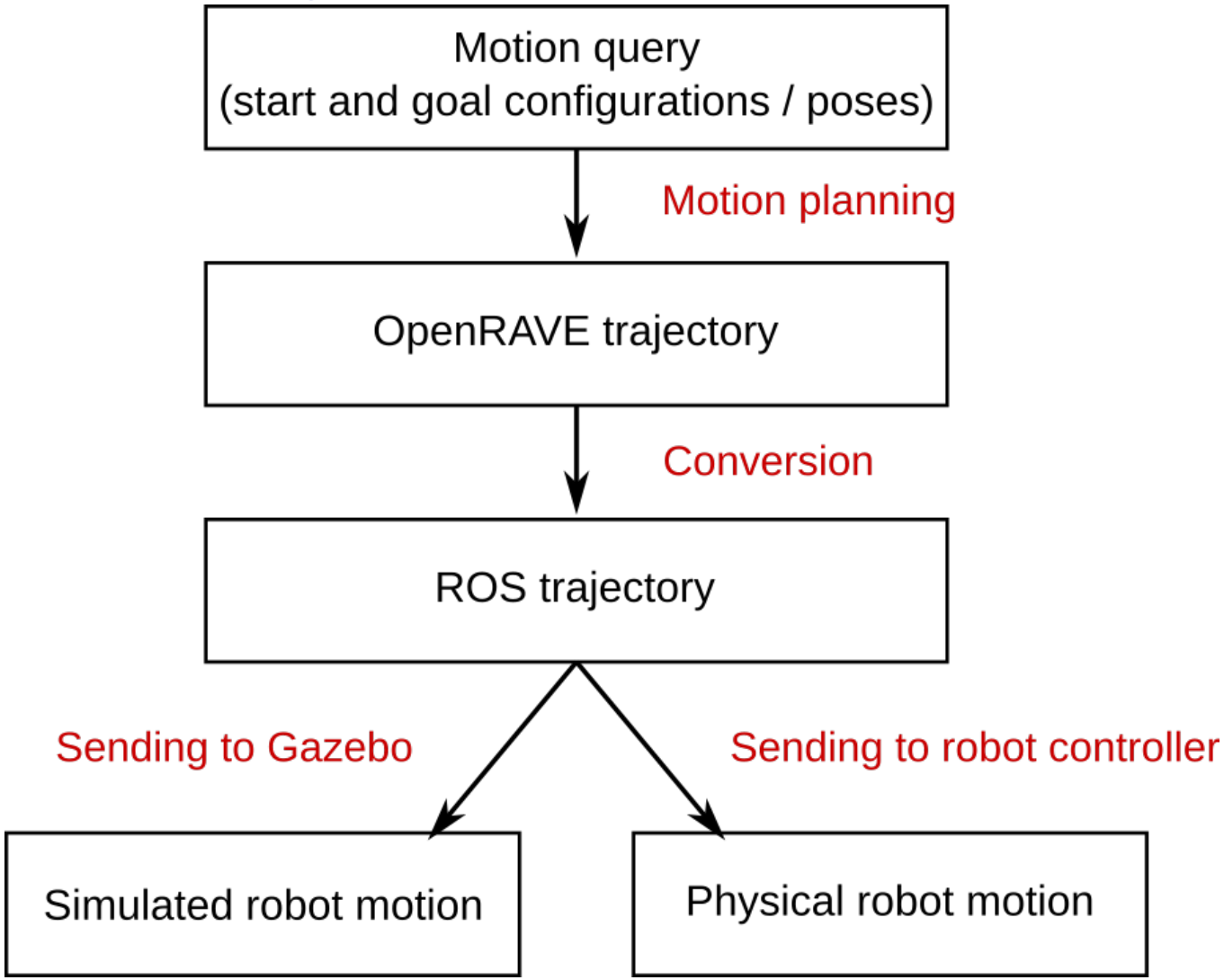
Main Gazebo Features

- **3D graphics.**
  Includes lighting, shadows, texture, etc.
- **Dynamic physics simulation.**
  Multiple physics engine available.
- **Various simulated, noisy sensors.**
  laser range finders, monocular & stereo cameras, Kinect, etc.
- **Numerous 3D robots.**
  Robots are models + physical characteristics.
  Several popular real robots have pre-made simulation counterparts.
  While tricky, possible to create own custom robots.
- **Plugin API**
  Can develop plugins that programatically affecting robots, sensors, or the environment.
- Numerous existing plugins provide extensive ROS support.

Gazebo


GAZEBO + ROS
Meta Package: gazebo_ros_pkgs
gazebo
Stand Alone Core
urdfdom
gazebo_msgs
Msg and Srv data structures for interacting with Gazebo from ROS.
gazebo_tests
Merged to gazebo_plugins
Contains a variety of unit tests for gazebo, tools and plugins.
gazebo_ros_api_plugin
Gazebo Subscribed Topics
~/set_link_state
~/set_model_state
Gazebo Published Parameters
/use_sim_time
Gazebo Published Topics
/clock
~/link_states
~/model_states
Gazebo Services
~/spawn_urdf_model
~/spawn_sdf_model
~/delete_model
State and properties getters
...
State and properties setters
...
Simulation control
~/pause_physics
~/unpause_physics
~/reset_simulation
~/reset_world
Force control
~/apply_body_wrench
~/apply_joint_effort
~/clear_joint_forces
~/clear_body_wrenches
gazebo_ros
Formally simulator_gazebo/gazebo
This package wraps gzserver and gzclient by using two Gazebo plugins that provide the necessary ROS interface for messages, services and dynamic reconfigure
ROS node name:
gazebo
Plugins:
gazebo_ros_api_plugin
gazebo_ros_paths_plugin
Usage:
rosrun gazebo_ros gazebo
rosrun gazebo_ros gzserver
rosrun gazebo_ros gzclient
rosrun gazebo_ros spawn_model
rosrun gazebo_ros perf
rosrun gazebo_ros debug
gazebo_plugins
Robot-independent Gazebo plugins.
Sensory
gazebo_ros_projector
gazebo_ros_p3d
gazebo_ros_imu
gazebo_ros_laser
gazebo_ros_f3d
gazebo_ros_camera_utils
gazebo_ros_depth_camera
gazebo_ros_openni_kinect
gazebo_ros_camera
gazebo_ros_bumper
gazebo_ros_block_laser
gazebo_ros_gpu_laser
Motory
gazebo_ros_joint_trajectory
gazebo_ros_diffdrive
gazebo_ros_force
gazebo_ros_template
Dynamic Reconfigure
vision_reconfigure
hokuyo_node
camera_synchronizer
gazebo_worlds
Merged to gazebo_ros
Contains a variety of unit tests for gazebo, tools and plugins.
wg
simple_erratic
simple_office
wg_collada_throttled - delete
wg_collada
grasp
empty_throttled
3stacks
elevator
simple_office_table
scan
empty
simple
balcony
camera
test_friction
simple_office2
empty_listener
gazebo_tools
Removed
gazebo_ros_paths_plugin
Provides ROS package paths to Gazebo
ROS packages
Gazebo Plugin
Depreciated from simulator_gazebo

Example pipeline using OpenRAVE motion planner



Getting Started Resources

VIDEO **Gazebo quick start, with Turtlebot**:
`www.youtube.com/watch?v=9U6GDonGFHw`

VIDEO **Robots and sensors from scratch**:
`www.youtube.com/watch?v=8ckSl4MbZLg`

- Basic tutorials (Turtlebot):
  `learn.turtlebot.com/2015/02/03/3/`
- **Gazebo tutorials**:
  `github.com/SMARTlab-Purdue/ros-tutorial-gazebo-simulation`
- **Use Fetch and Freight robots**:
  `docs.fetchrobotics.com/gazebo.html`
- **Modify Gazebo environment through ROS topics**:
  `gazebosim.org/tutorials/?tut=ros_comm`
- **Gazebo + ROS + MATLAB**:
  `mathworks.com/help/robotics/examples/get-started-with-gazebo-and-a-simulated-turtlebot.html`
  Their examples avoid plugins by working with ROS topics. Includes prebuilt virtual machine with Turtlebot for easy learning.

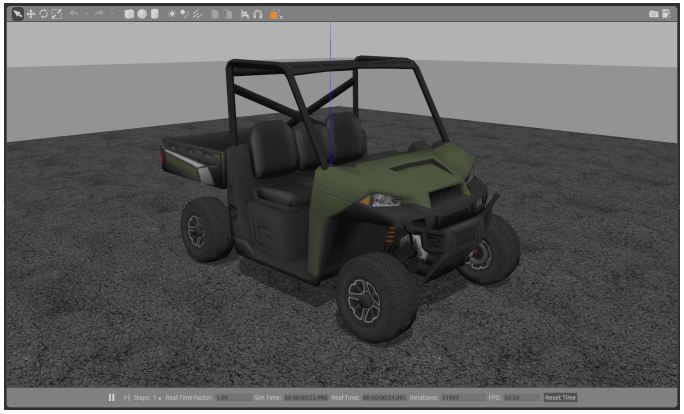
Challenges

Gazebo offers a wealth of resources. Many projects can be done by combining existing tools without too much trouble. However, custom functionality requires a rather steep learning curve.

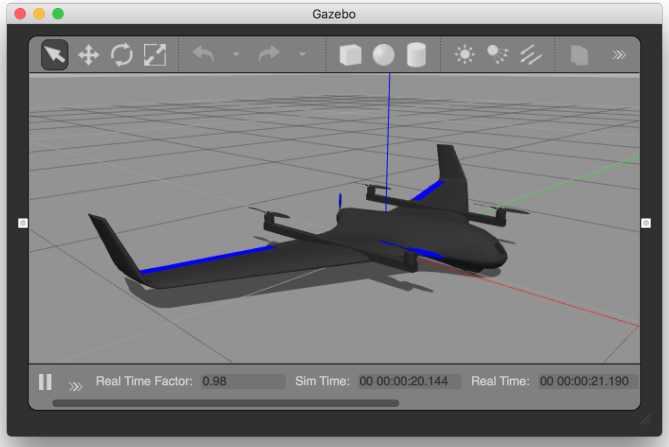
- Documentation is extensive, but non-trivial to navigate.
- Many significant plugins have minimal comments.
- Examples for using existing plugins might be sparse.
- Seemingly routine tasks requires writing plugins in Gazebo's verbose, complex API.
- Compiling plugins typically requires working with ROS's intricate catkin build system.
- Some confusion exists between Gazebo topics and ROS topics, with overlap.
  Gazebo can exist outside of ROS, so it has own commands for pub/sub to topics.
  But plugins might be running that pushes some or all of the Gazebo topics to ROS topics.

Example

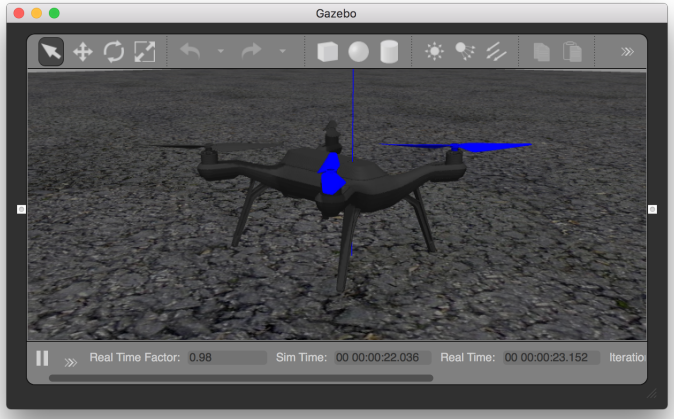
Rover



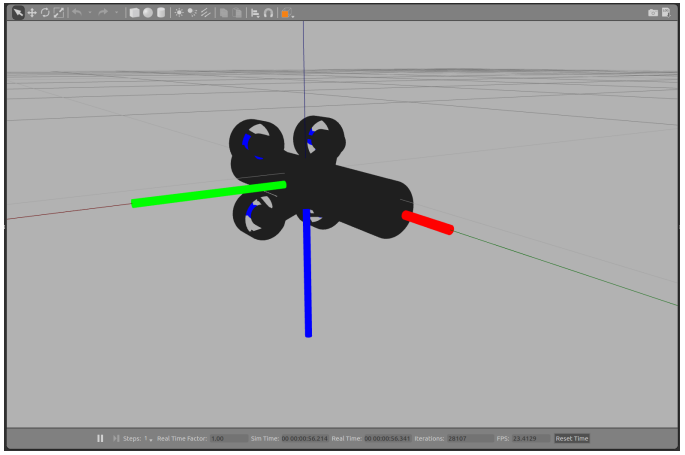
UAV



Quadcopter



Unmanned Underwater Vehicle



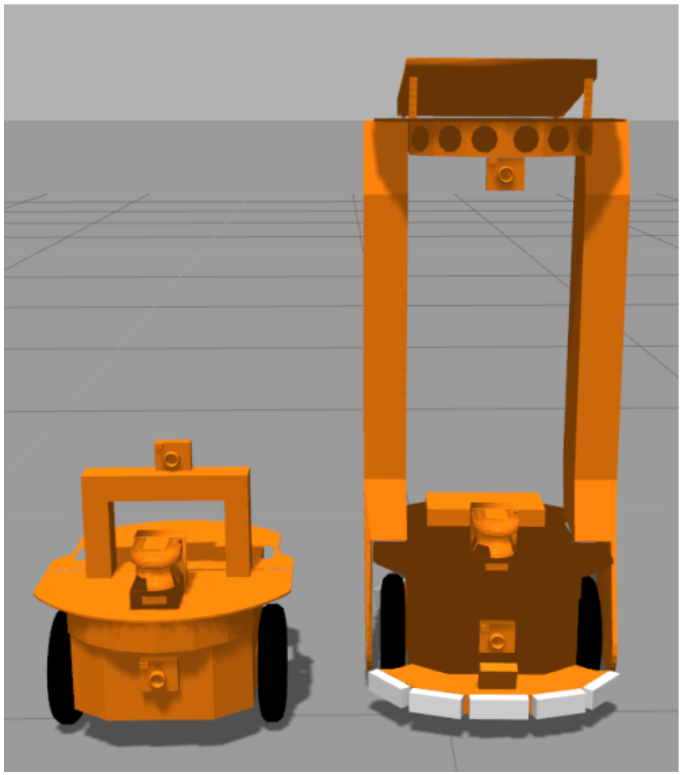
Example

Laser Scanner

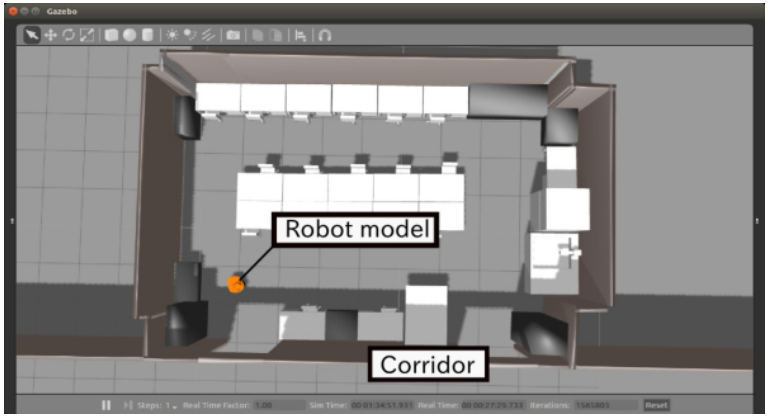
Example

3D Map Generation

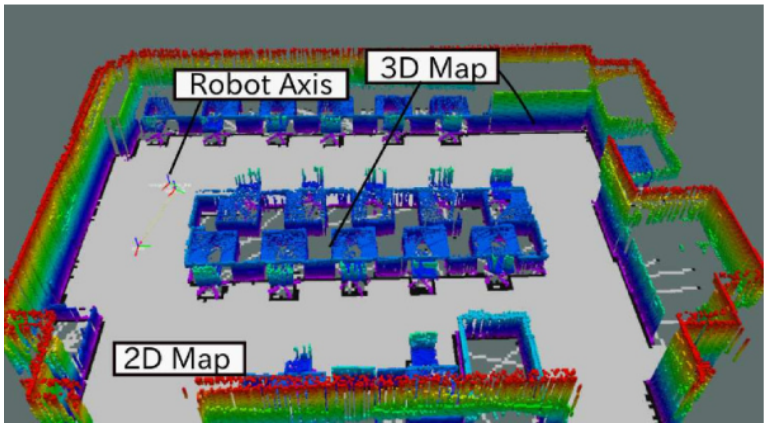
Two Robot Team



Gazebo Environment



Mapping Visualization



Example

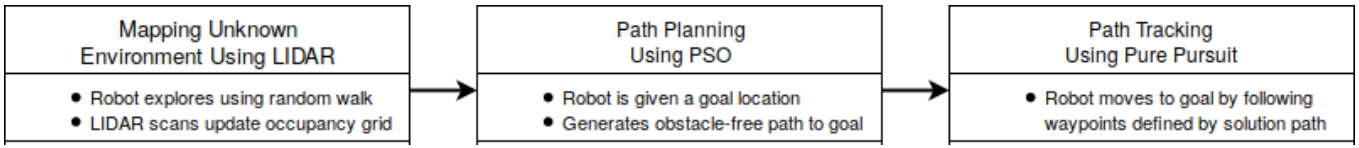
Path Planning using Particle Swarm Optimization

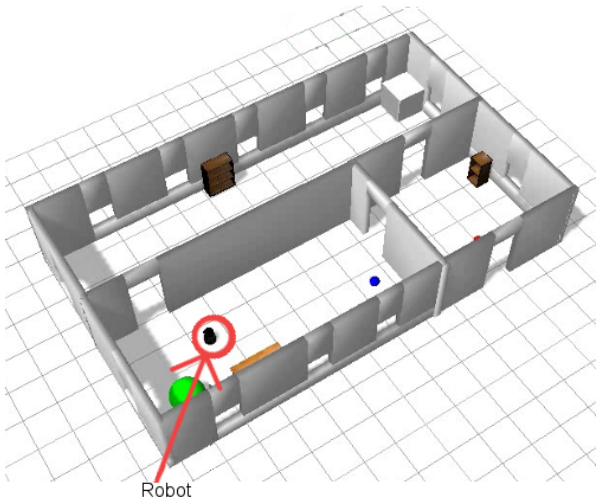
Collision-Free Autonomous Robot Navigation System Utilizing PSO for Path Planning in Unknown Environment

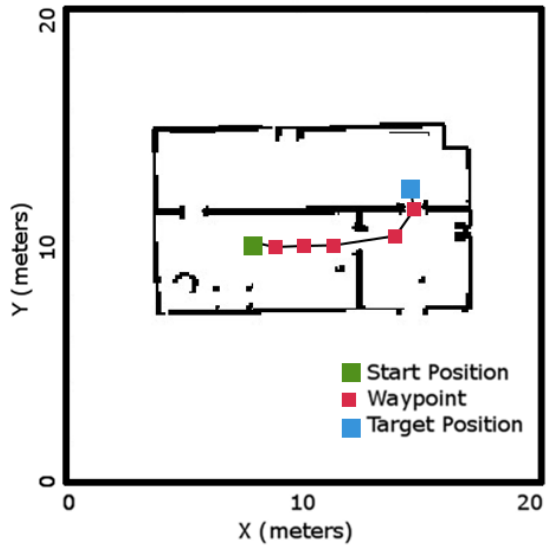
Evan Krell, Alaa Sheta, Arun Prassanth Ramaswamy Balasubramanian, Scott A. King

Accepted in *Journal of Artificial Intelligence and Soft Computing Research* (JAISCR)

`https://github.com/ekrell/RobotPathPlanningPSO`



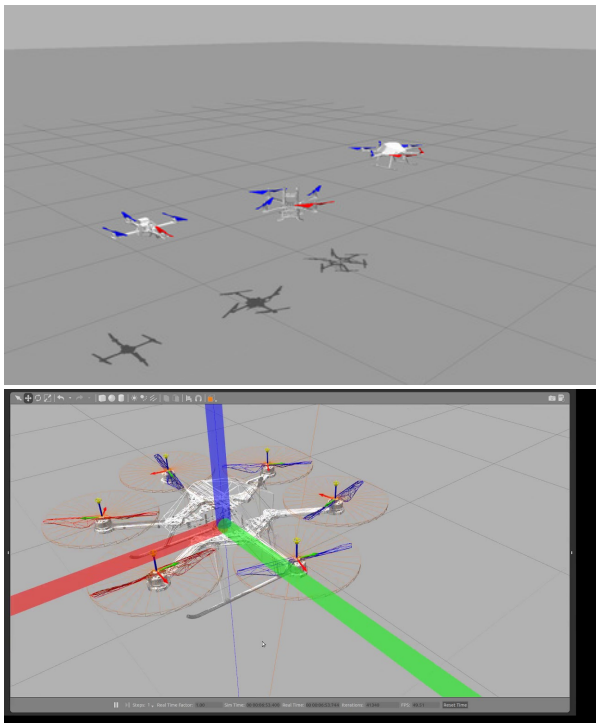




Tutorial: Create Robot and Sensor

- We will watch the video tutorial, but pausing to analyze the code.
- Video: `www.youtube.com/watch?v=8ckSl4MbZLg`
- Code: `github.com/richardw05/mybot_ws`

Multirotor Simulation with RotorS



`www.kostasalexis.com/rotors-simulator1.html`

RotorS

- Multirotor simulation
- Three *AscTec* vehicles: Hummingbird, Pelican, Firefly.
- Can add custom aircraft.
- Equipped with sensors: IMU, odometry, and Visual-Inertial (VI-) Sensor.
- Includes waypoint-following (autopilot) and hovering packages.
- Can build and test custom controllers.
- Comes with a joystick interface.
- Minimal tutorials & less documented than Turtlebot and other beginner-friendly packages.

Image Sources

osrobotics.org/osr/system/simulation.html
https://dev.px4.io/en/simulation/gazebo.html
http://blog.pal-robotics.com/
ros-simulation-available-for-pmb-2-tiagos-mobile-base/
Takaya,Kentaetal.\begingroup\let\relax\relax\
endgroup[Pleaseinsert\PrerenderUnicode{œ}intopreamble]
SimulationenvironmentformobilerobotstestingusingROSandGazebo.
\begingroup\let\relax\relax\endgroup[Pleaseinsert\
PrerenderUnicode{}intopreamble]
201620thInternationalConferenceonSystemTheory,
ControlandComputing(ICSTCC)(2016):96-101.
https://github.com/ethz-asl/rotors_simulator