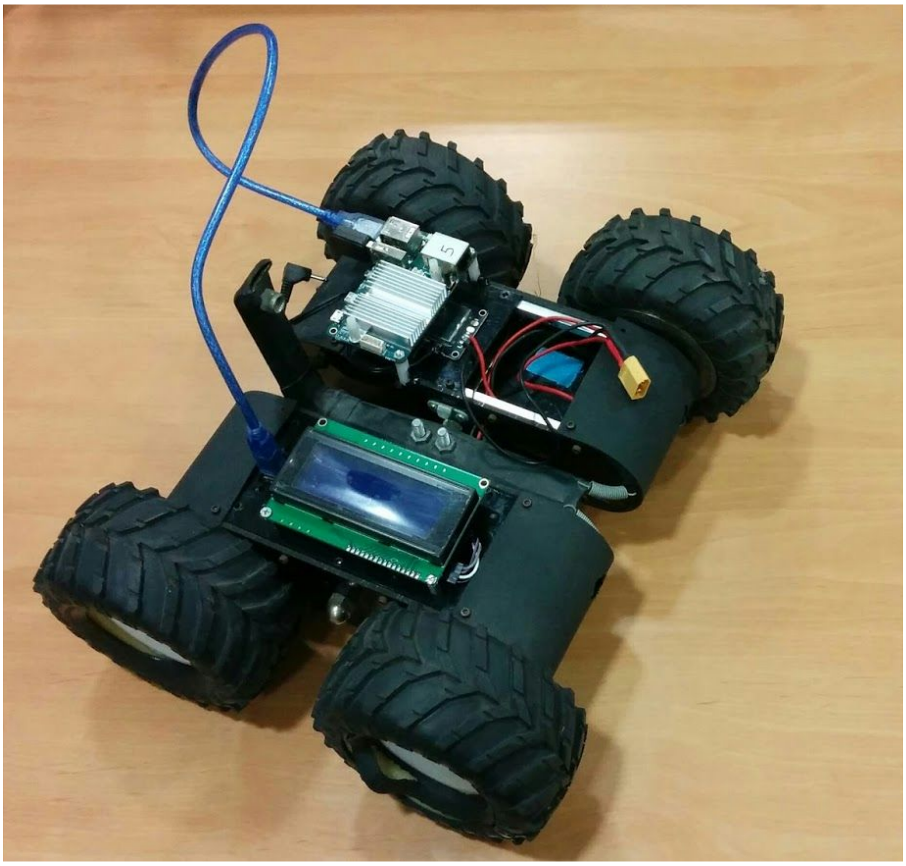
TRSA 2015

Final Projects

LinoBot

Skid Steer Locomotion of a small Autonomous Ground Vehicle



The mobility of any autonomous robot depends primarily on the way it is able to control its locomotion. This implies the definition of the commands relayed to the motors, and the locomotion modes made possible by the vehicle's dynamics.

Furthermore, the use of feedback information, from either the actual motor encoders, or any other sensor, to measure the actual movement produced by any given motor command is very important. The vehicle's localization will depend exclusively on the wheel odometry to position itself in the world, when no other sensor cue is available.

In this work you will have a four-wheeled robot, controlling its motor via an Arduino, connected to a Odroid running xubuntu. Your job is to implement the locomotion nodes responsible for the low-level communication with the motors and their encoders. Initially, you will have to program the routines in the Arduino that will control the motors and retrieve information of the encoders (***sensors_msgs/JointState***). The arduino will then be connected via serial port to the Odroid where a ROS node will have to receive command velocities (***geometry_msgs/Twist***) and publish odometry (***nav_msgs/Odometry***) information for high-level navigation. To interface the arduino with your ROS network please see the following tutorial *http://wiki.ros.org/rosserial_arduino/Tutorials*. There are several packages that implement skid-steer locomotion.

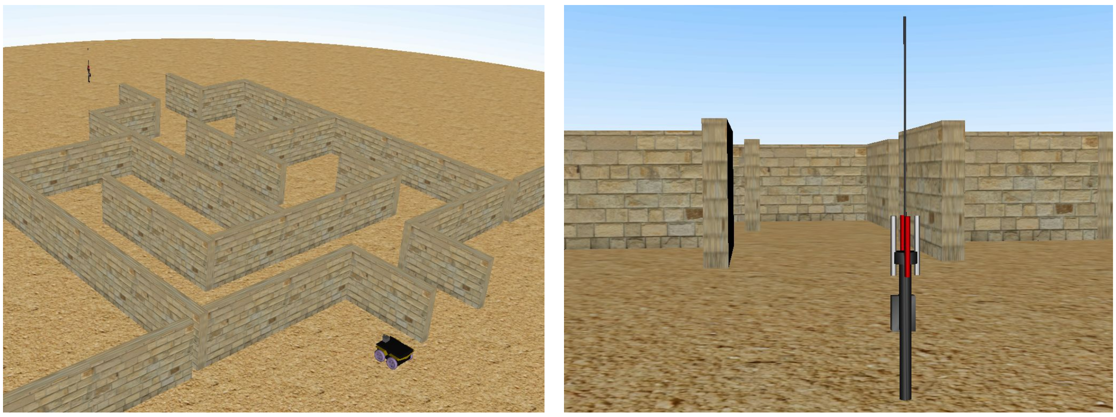
REQUIREMENTS: LINObot, Sample arduino program.

Maze Runner

Autonomous exploration of unknown environments

A serious problem comes from how to optimize an exploration of an unknown and heterogeneous environment. This application is especially important in search and rescue operations. When faced with an unknown setting the robot should be able to gather enough information to reach its objective in the least amount of time possible.

In this work the robot's objective is represented by a transponder that sends GPS beacon messages with its location. The platform that you control is omnidirectional equipped with a 2D laser scanner, IMU, Odometry and GPS. The virtual environment is a maze created with brick walls, to be able to traverse you will have to devise an exploration strategy that maximizes the information gathering as well as trying to localize and reach the final objective represented by the transponder. One possible application of such a concept is the localization of a plane's black box.



(a) (b)

Fig. 1. The simulated environment. (a) A aerial perspective of the maze the robot has to navigate to reach its goal. (b) The GPS beacon the robot has to reach.

There have several ROS modules that already have basic functionalities for robot navigation and exploration. As a suggestion, the student should start by exploring the ROS navigation package: http://wiki.ros.org/navigation.

REQUIREMENTS: Kelp simulator (RICS Research Group)

BIBLIOGRAPHY:

Thrun, Sebastian, et al. "Autonomous exploration and mapping of abandoned mines." *Robotics & Automation Magazine, IEEE* 11.4 (2004): 79-91.

AVAILABLE ROS MESSAGE TOPICS:

/kelp/gps

/kelp/gps_beacon

/kelp/imu

/kelp/laser_scan

/kelp/laser_scan/joint_state

/kelp/odometry

/kelp/robot_control

AVAILABLE ROS SERVICE TOPICS:

/kelp/laser_scan/servo_control

Note: Kelp simulator can (and should be) run in a dedicated machine and all other ROS packages in another machine for the overall performance.

Please see the following tutorial: http://wiki.ros.org/ROS/Tutorials/MultipleMachines

Sea Explorer

Sonar Mapping for Autonomous Surface Navigation

Considering an autonomous **aquatic-surface** vehicle's underwater perception, a sonar range profiler is the most common approach. Usually, the sonar is mounted facing down for bathymetry purposes. However, to improve the robot's look-ahead underwater, it may be required that the scanner is directed, or tilted, forward. Thus objects in the robot's passageway, precisely the ones that may collide with the underwater part of the vessel, may be properly detected. Doing that may impair a good reconstruction of the bottom of the river or lake. As Fig 2 depicts, irregular terrain underneath, hills and such, may occlude the sonar at times.

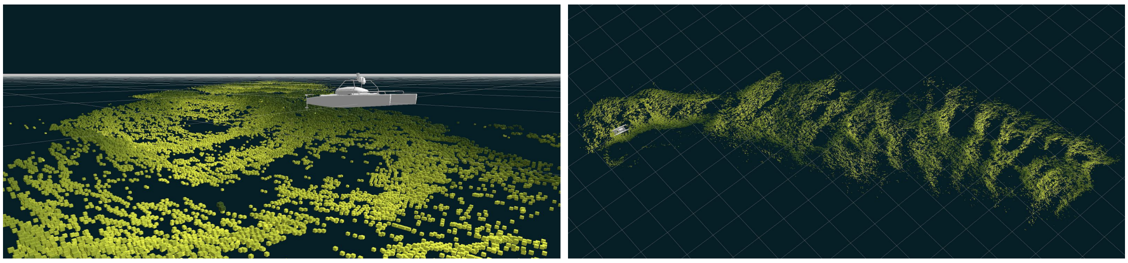
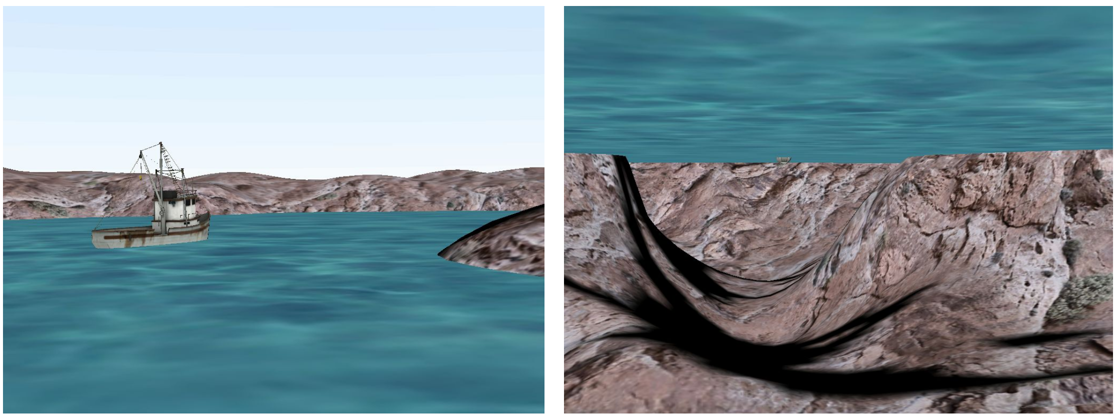


Fig. 2 – Gaps in Bathymetry caused by a fixed 21° tilt forward of the sonar.

Your work will be to first identify the gaps in the 3-D cloud, reconstruct the surface and derive a 2.5D map. This 2.5D map will resolve which areas surrounding the ASV are navigable and which present obstacles. Besides that, it will act as service provider to other vessels, which lack this sensory input to derive their own obstacle maps. In this sense, a service should be implemented so other vessels may call upon, providing their underwater vertical clearance, to obtain a dedicated 2-D obstacle map (***nav_msgs/OccupancyGrid*** or ***nav_msgs/GridCells***).



(a) (b)

Fig. 3 – Proposed virtual world for the sea explorer project.

Finally, while reconstructing the bottom surface solves the problem for water surface navigation, it may be required of the ASV to harness a complete bathymetric image of the terrain, to be employed, for instance, on sub-surface autonomous systems. To that end, the ASV should be directed, in the form of ***geometry_msgs/TwistStamped*** - like message, to revisit those regions lacking proper sampling.

REQUIREMENTS: Kelp simulator (RICS Research Group)

AVAILABLE ROS MESSAGE TOPICS:

/kelp/gps

/kelp/imu

/kelp/laser_scan

/kelp/laser_scan/joint_state

AVAILABLE ROS SERVICE TOPICS:

/kelp/laser_scan/servo_control

Note: Kelp simulator can (and should be) run in a dedicated machine and all other ROS packages in another machine for the overall performance.

Please see the following tutorial: http://wiki.ros.org/ROS/Tutorials/MultipleMachines

Auto Landing

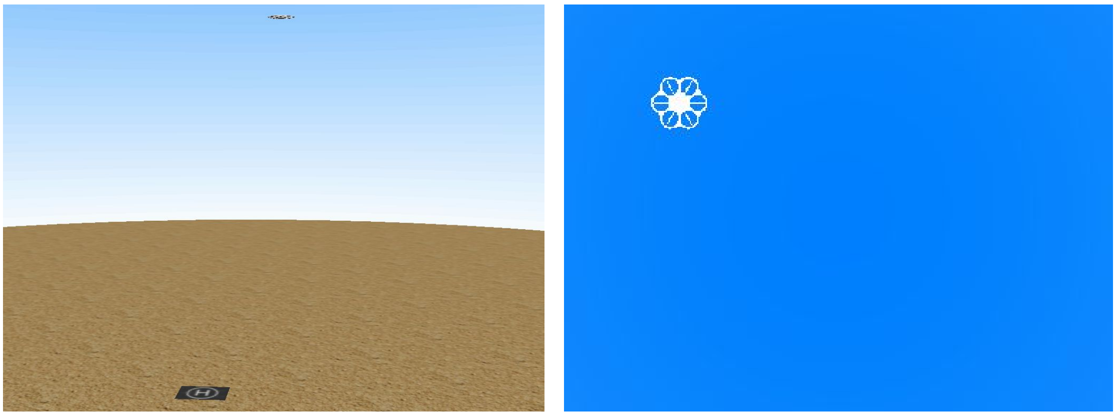
Active perception based in multiple cues

In order to optimize the sensory acquisition, it would be important to redirect the sensory equipment to the vantage point most suited to collect the maximum amount of information from the environment. If something of interest is picked up by any of the sensors, all sensors would be best employed if they focused/centered their field-of-views upon that something.

Consider now the robotic team composed by the quadcopter and the boat displayed on Fig. 4. The aerial vehicle has the ability to take-off and land on a dedicated platform, a helipad, aboard the surface vehicle. To do that, the UAV takes advantage of both a camera directed down to its landing zone and another mounted on the ASV's helipad, facing upwards. To reinforce the UAV's localization in respect to the helipad, it would be interesting to have the ASV's laser actively track the UAV's position. Hence, an attention mechanism is, fed by the upwards camera system's image location and focal length, would drive the tilting servo to align the laser scanner to sense the UAV.

Fig. 4. The ASV-UAV marsupial robotic team. The UAV just took off the ASV in order to engage on an aerial survey for expanding the robotic system's awareness of the environment. As a result of this cooperative perception, the robotic ensemble is capable of setting navigation plans with a lookahead that far exceeds the one available from the ASV's onboard sensors.

The student's work is to detect the aerial vehicle's position on the upwards camera field-of-view (see Fig 5(b)), and, having the calibrated camera's focal length, estimate the UAV's 3-D position in respect to the camera's reference frame. Then, evaluating the transforms of the ASV, your system should be able to determine the laser scanner's tilt angle most likely directing it to UAV's. The final product of your work will thus be a transform broadcasting the UAV's position relative to the world reference frame. The Laser scanner position is X: 1 m, Y: 1m, Z: 1m from the centre of the helipad. You should control the angle of the tilt via the appropriate service.



(a) (b)

Fig. 5 – Proposed virtual world for the auto landing project. (a) The UAV approaching the helipad. (b) The UAV seen through the Helipad camera.

REQUIREMENTS: Kelp simulator (RICS Research Group)

AVAILABLE ROS MESSAGE TOPICS:

/kelp/helipad/camera/image

/kelp/helipad/camera/focal_length

/kelp/helipad/gps

/kelp/robot_control

/kelp/uav/gps

/kelp/uav/imu

/kelp/uav/laser_scan

/kelp/uav/laser_scan/joint_state

AVAILABLE ROS SERVICE TOPICS:

/kelp/uav/laser_scan/servo_control

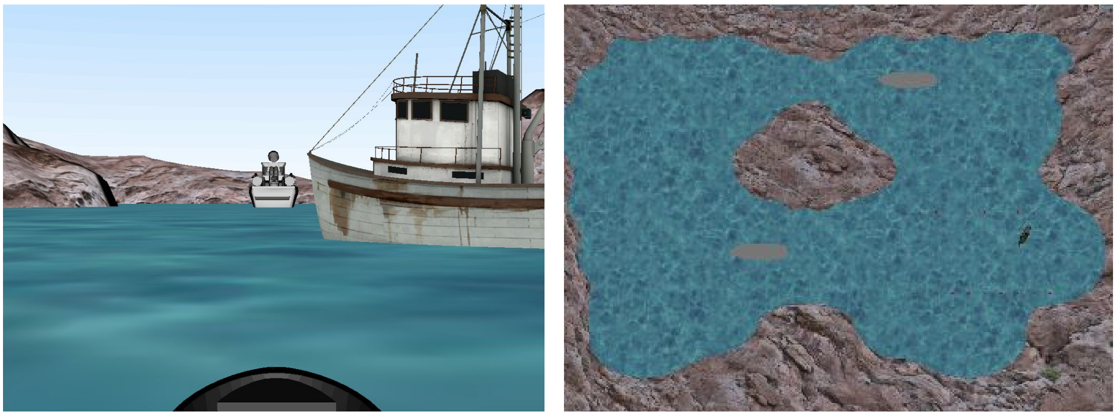
Note: Kelp simulator can (and should be) run in a dedicated machine and all other ROS packages in another machine for the overall performance.

Please see the following tutorial: http://wiki.ros.org/ROS/Tutorials/MultipleMachines

Sea Navigator

Basic implementation of COLREGS

To ensure safe navigation of navigating vessels COLREGs are derived from a multilateral treaty called the **Convention on the International Regulations for Preventing Collisions at Sea** COLREG . The objective of this work is to implement these conventions. You will have access to the current position of each boat as well as the most probable path of each vessel. According to this information you should control your vessel to obey to the appropriate COLREG (see Fig 6).



(a) (b)

Fig. 6 - (a) Depiction of the application of a COLREG. In this example the vessel moving from the right side has priority. With access to an estimation the vessel's path your vessel needs to replan its motion to account to the respective COLREG. (b) Example of the aerial view provided by the simulated UAV to aid the detection of other vessels

Your ASV will be equipped with a laser scanner, GPS and IMU that can be used to estimate its own position and of other objects.

You will be able to access images retrieved by a aerial vehicle flying overhead to help the detection and motion estimation of other entities throughout the environment (see Fig. 1). Tracking the motion of dynamic objects would allow, for instance, a good estimation of which colreg needs to be applied. The position of the UAV, and the camera itself, will be retrieved from GPS and altitude measurements. That way it will be possible to detect and track the objects from the camera's point of view, and render all objects GPS coordinates.

Initially, you should exploit the fact that all vessels publish their GPS position. However, the objective of this work is to determine the correct COLREG without this information. The laser scanner on the ASV X:1m, Y:0m, Z:0.5m.

REQUIREMENTS: Kelp simulator (RICS Research Group)

BIBLIOGRAPHY:

Perera, L. P. (2010). Autonomous guidance and navigation based on the COLREGs rules and regulations of collision avoidance, (1999), 205–216.

AVAILABLE ROS MESSAGE TOPICS:

/kelp/uav/camera/image

/kelp/uav/camera/focal_length

/kelp/uav/gps

/kelp/asv/gps

/kelp/asv/imu

/kelp/asv/laser_scan

/kelp/asv/laser_scan/joint_state

/kelp/asv/odometry

/kelp/boat_1/gps

/kelp/boat_2/gps

/kelp/boat_3/gps

/kelp/robot_control

AVAILABLE ROS SERVICE TOPICS:

/kelp/asv/laser_scan/servo_control

Note: Kelp simulator can (and should be) run in a dedicated machine and all other ROS packages in another machine for the overall performance.

Please see the following tutorial: http://wiki.ros.org/ROS/Tutorials/MultipleMachines