

Field Robotics and Learning Group, Mechanical Engineering, University College London

Yanchao Wang, Song Ma, Xu You, Zaopeng Dong, Qianqian Yang, Yongchang Xie, Yuanchang Liu  
yanchao.wang.20@ucl.ac.uk, song.ma.18@ucl.ac.uk, yuanchang.liu@ucl.ac.uk



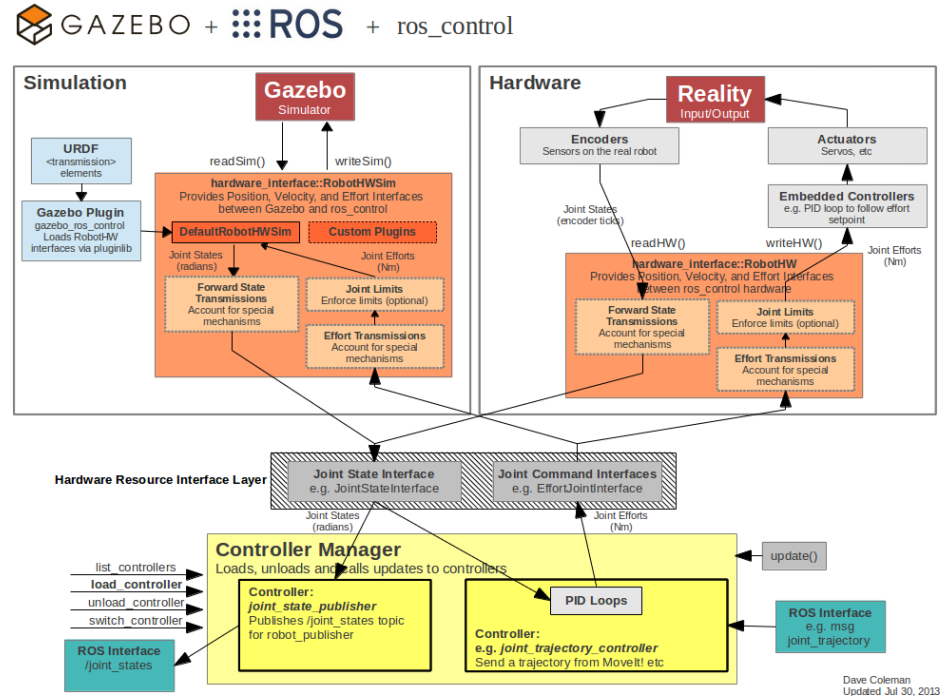
# Fostering Skills and Expertise in Marine Robotics at UCL: Virtual RobotX 2023

Team FRL



# Robot Operating System (ROS) and Gazebo

- Hardware-in-the-loop (HIL) is a testing technique used in the development of complex systems, particularly in engineering disciplines like robotics and control systems. In HIL testing, the physical hardware components of a system, such as sensors, actuators, and controllers, are integrated into a simulation environment.



Dave Coleman  
Updated Jul 30, 2013

# About Virtual RobotX 2023

## Individual Tasks

The following quick start instructions walk you through the available task-specific messages.

Task 1: [Stationkeeping](#)

Task 2: [Wayfinding](#)

Task 3: [Perception](#)

Task 4: [Acoustic Perception](#)

Task 5: [Wildlife Encounter and Avoid](#)

Task 6: [Follow the Path](#)

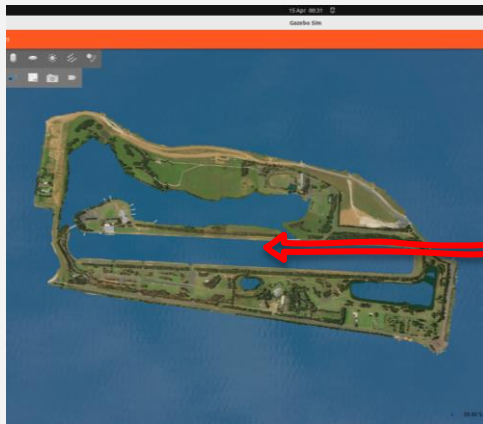
Task 7: [Acoustic Tracking](#)

Task 8: [Scan and Dock and Deliver](#)

Top: [VRX Tutorials](#)

Next: [Stationkeeping](#)

The VRX simulation environment has been embraced by the maritime robotics research and development community, evolving to accommodate advancements in perception, learning, and control, along with exploring novel applications for USV capabilities.



<https://robotx.org/programs/vrx-2023/>

# Customizing WAM-V

- The WAM-V thruster and component configuration can be customized. This involves writing a user-generated thruster YAML file and a user-generated component YAML file, and then running a script that will generate a custom WAM-V URDF file with the specified thrusters and components.



# Station Keeping

- **NMPC:**

- The USV model can be denoted as:

$$\begin{bmatrix} \mathbf{m} & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} \dot{v}_W \\ \dot{\omega}_B \end{bmatrix} = \begin{bmatrix} \dot{F}_W \\ \dot{\tau}_B \end{bmatrix} - \begin{bmatrix} \omega_B \times m v_B \\ \omega_B \times I \omega_B \end{bmatrix} + \begin{bmatrix} m g_W \\ 0 \end{bmatrix}$$

- The cost function to minimize the control effort:

$$J(x_t, u_t) = \sum_{t=k}^{k+N-1} [(x_t - x_t^r)^\top Q (x_t - x_t^r) + u_t^\top R u_t] \\ + (x_{k+N} - x_{k+N}^r)^\top F (x_{k+N} - x_{k+N}^r)$$

- The constrained optimal control problem in the prediction horizon is sampled in N steps. At each time t, the problem can be denoted as:

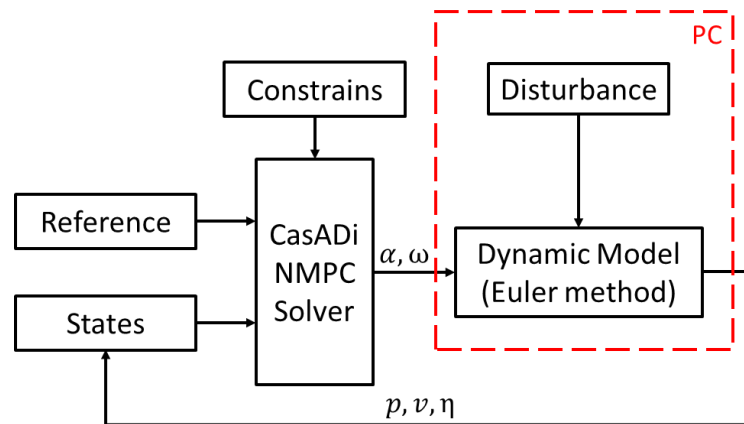
$$\min_{x_0 \dots x_N, u_0 \dots u_{N-1}} J$$

subject to

$$x_{k+1} = f(x_k, u_k)$$

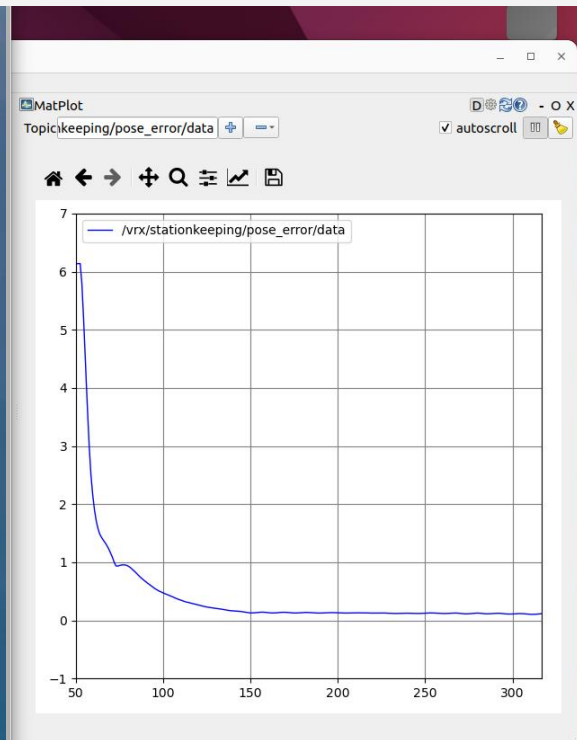
$$x_k \in \mathbb{X}$$

$$u_k \in \mathbb{U}$$



**Task Summary:** Navigate to the goal pose and hold station. The best solutions will minimize the difference between the goal pose and the actual pose of the vehicle over the duration of the task.

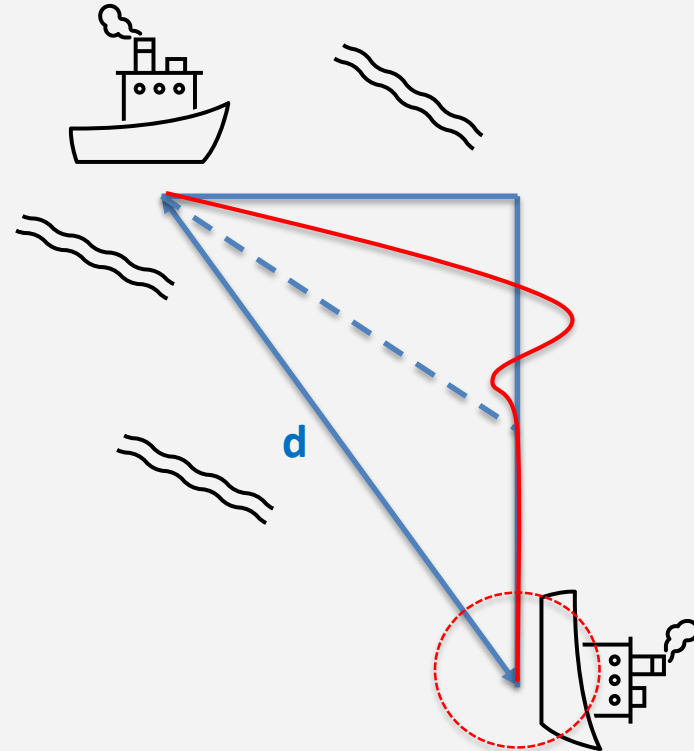
# Station Keeping



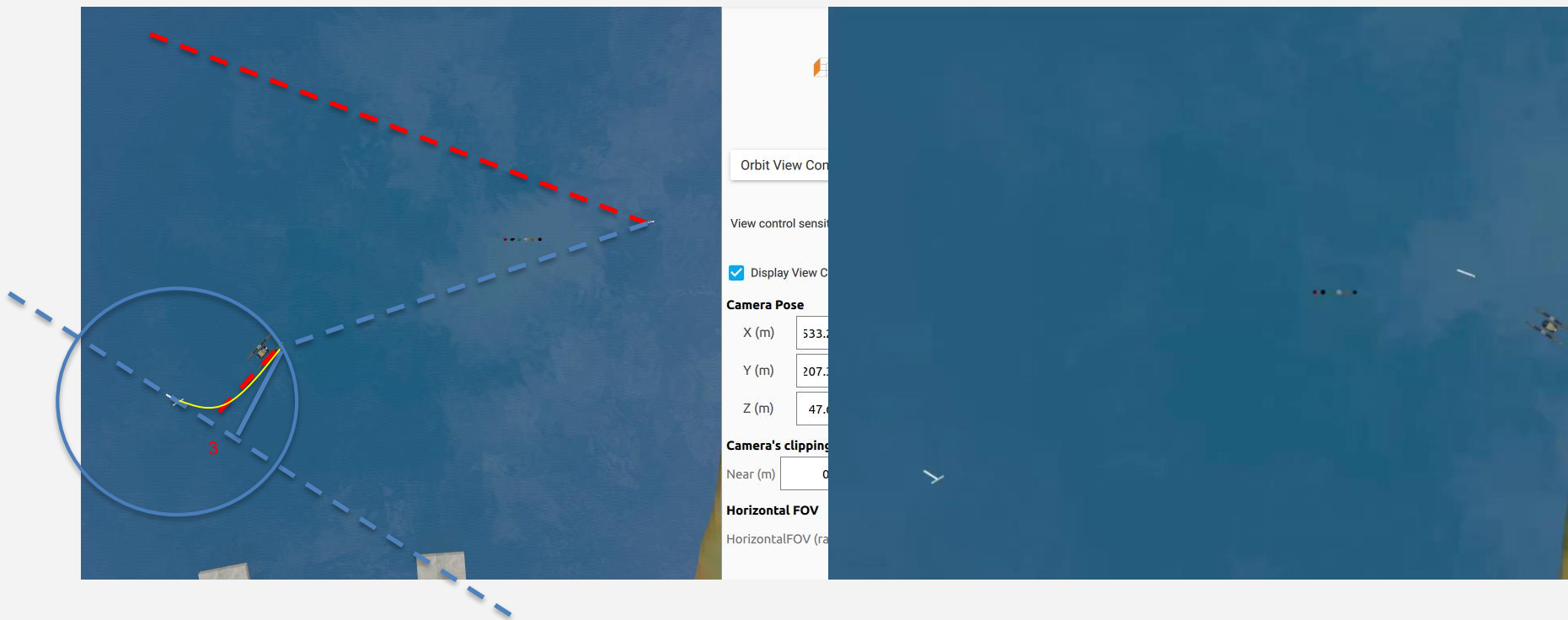


# Waypoint Tracking

- **Task Summary:** Navigate through each of the published waypoints, such that vehicle achieves, as closely as possible, the positions and orientations specified.
- **Solution:**
  - When the distance is longer than the desired threshold, the WAM-V will following a desired trajectory.
  - When the distance is shorter than the desired threshold, the WAM-V will start the station keeping procedure.
- **Benefits:**
  - Gazebo provides users with a convenient testing method when their models are difficult to manipulate or have safety risks.



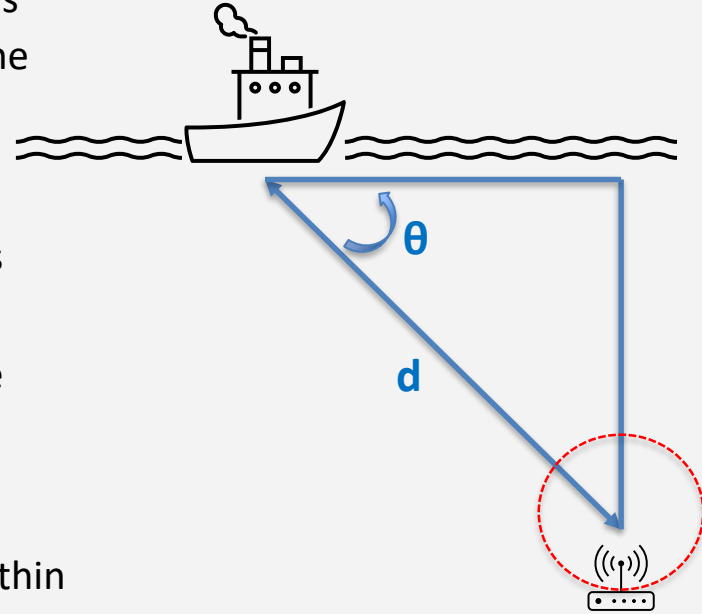
# Waypoint Tracking





# Acoustic Perception

- **Task Summary:** An underwater acoustic beacon emits noisy signals representing its position relative to the USV. The objective is for the USV to locate and navigate towards.
- **Solution:**
  - Use a Butterworth lowpass filter to process the noisy signals broadcasted by the beacon.
  - Calculate desired heading and speed for the USV to navigate towards the beacon dynamically
- **Benefits:**
  - Gazebo can simulate a variety of disturbances and noises within the open-source environment, enhancing the realism of sensor noise in real-world scenarios.



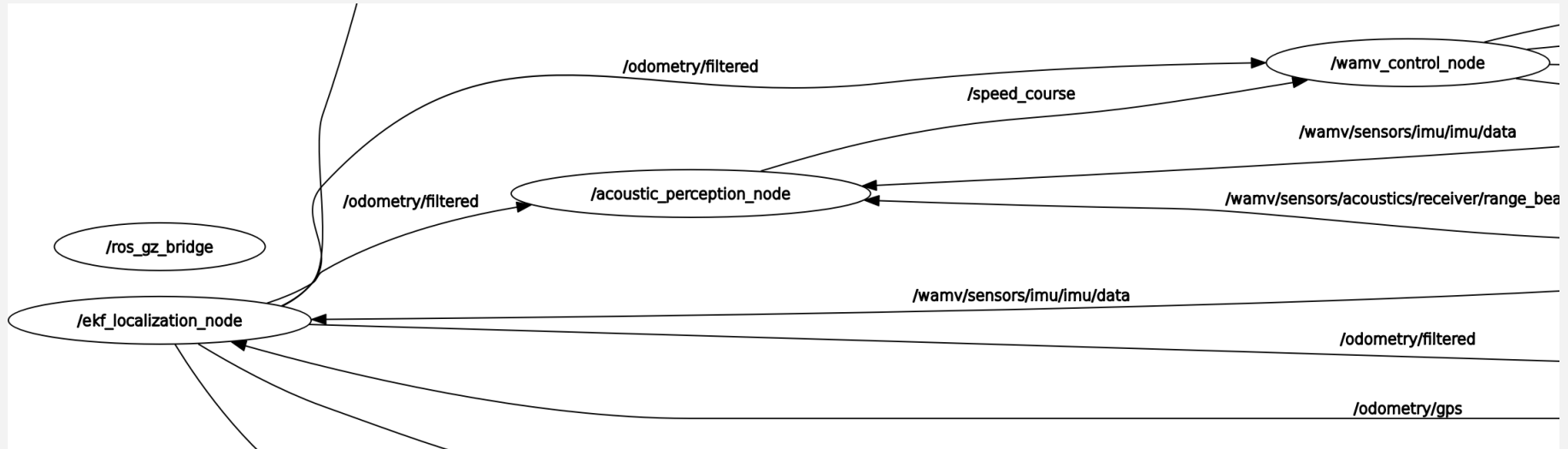
# Acoustic Perception



```
yanchao@yanchao-Latitude-5420: ~/yanchao_vrx_ws/src/vrx_...  
yao@yanchao-Lat... x yanchao@yanchao-Lat... x yanchao@yanchao-Lat... x  
Position Error: 14.730  
coustic_perception-3] [INFO] [1713132932.745740518] [acoustic_perception_n  
Position Error: 14.730  
coustic_perception-3] [INFO] [1713132932.945765541] [acoustic_perception_n  
Position Error: 14.730  
coustic_perception-3] [INFO] [1713132933.145754493] [acoustic_perception_n  
Position Error: 14.730  
coustic_perception-3] [INFO] [1713132933.345734562] [acoustic_perception_n  
Position Error: 14.730  
coustic_perception-3] [INFO] [1713132933.545996898] [acoustic_perception_n  
Position Error: 14.730  
coustic_perception-3] [INFO] [1713132933.745749684] [acoustic_perception_n  
Position Error: 14.427  
coustic_perception-3] [INFO] [1713132933.945811566] [acoustic_perception_n  
Position Error: 14.315  
coustic_perception-3] [INFO] [1713132934.145718967] [acoustic_perception_n  
Position Error: 14.259  
coustic_perception-3] [INFO] [1713132934.345760163] [acoustic_perception_n  
Position Error: 14.259  
coustic_perception-3] [INFO] [1713132934.545742058] [acoustic_perception_n  
Position Error: 14.259  
coustic_perception-3] [INFO] [1713132934.745767924] [acoustic_perception_n  
Position Error: 14.259
```

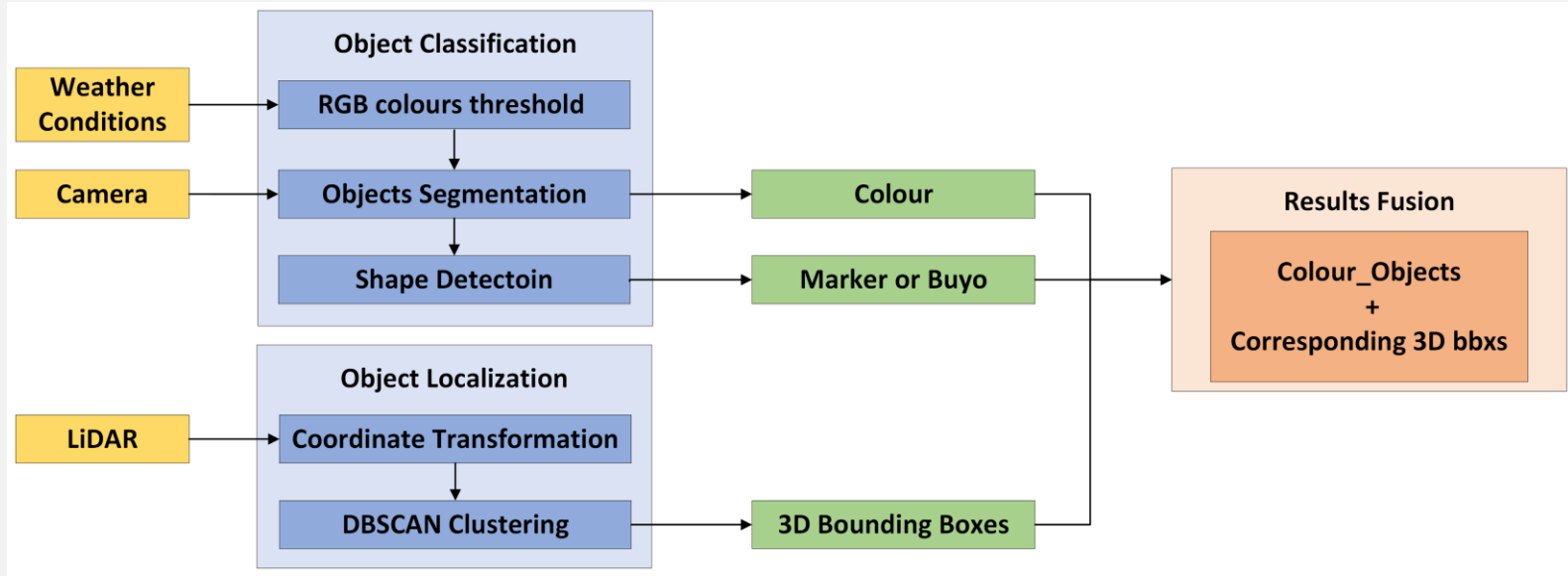
# Acoustic Perception

Nodes are typically used for various tasks such as controlling hardware, processing sensor data, performing calculations, or implementing algorithms. Nodes communicate with each other by publishing and subscribing to topics, which allows them to exchange data and coordinate their actions within a ROS system.



# Perception

- Task Summary:** The USV needs to utilize its sensors like cameras and LIDAR to identify, analyze, and pinpoint various objects in its surroundings. The perception process must be reliable enough to tackle uncertainties related to the vehicle's movement, sensor uncertainty, and environmental variations such as lighting changes and fog.



# Perception

The screenshot displays a ROS2 environment with several windows. The main window is a Gazebo simulation of a boat on a lake. A red jagged line highlights a small green object in the water, labeled "detected\_object". A "Detected Object" window shows a screenshot of this object. A "Message Publisher" window is open, showing a table of topics and their rates. A red box highlights the topics `/wamv/thrusters/left/thrust` and `/wamv/thrusters/right/thrust`.

Detected Object

You can paste the image from the clipboard.

Default - rqt

Topic Monitor

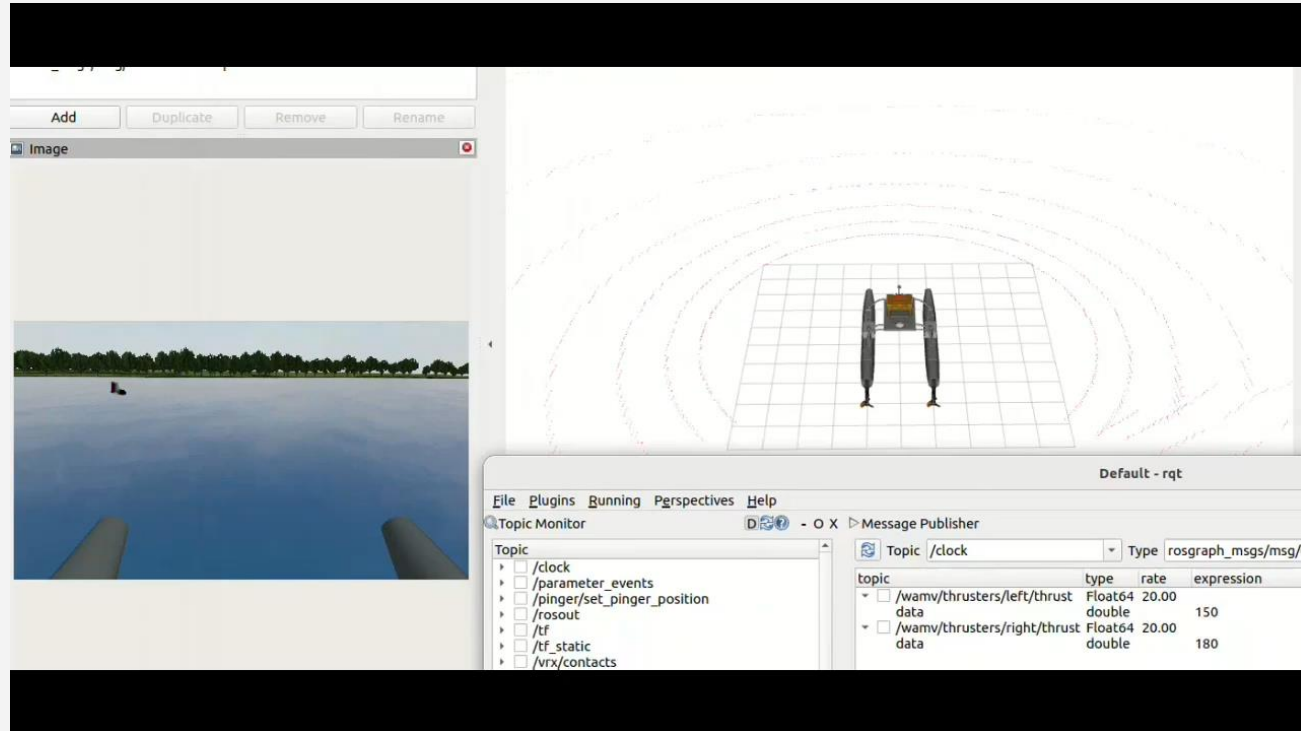
Topic	Type
/clock	roscpp_msgs/msg/Clock
/parameter_events	rcd_interfaces/msg/ParameterEvent
/ping/set_ping_position	geometry_msgs/msg/Vector3
/rosout	rcd_interfaces/msg/Log
/tf	tf2_msgs/msg/TFMessage
/tf_static	tf2_msgs/msg/TFMessage
/vrx/contacts	ros_gz_interfaces/msg/Contacts
/vrx/debug/wind/direction	std_msgs/msg/Float32
/vrx/debug/wind/speed	std_msgs/msg/TFMessage
/vrx/perception/landmark	geometry_msgs/msg/PoseStamped
/vrx/task_info	ros_gz_interfaces/msg/ParamVec
/wamv/joint_states	sensor_msgs/msg/JointState
/wamv/pose	tf2_msgs/msg/TFMessage
/wamv/pose_static	std_msgs/msg/TFMessage
/wamv/robot_description	std_msgs/msg/String
/wamv/sensors/acoustics/receiver/range_bearing	ros_gz_interfaces/msg/ParamVec
/wamv/sensors/cameras/front_left_camera_sensor/camera_info	sensor_msgs/msg/CameraInfo
/wamv/sensors/cameras/front_left_camera_sensor/image_raw	sensor_msgs/msg/Image
/wamv/sensors/cameras/front_left_camera_sensor/optical/camera_info	sensor_msgs/msg/CameraInfo
/wamv/sensors/cameras/front_left_camera_sensor/optical/image_raw	sensor_msgs/msg/Image
/wamv/sensors/cameras/front_right_camera_sensor/camera_info	sensor_msgs/msg/CameraInfo
/wamv/sensors/cameras/front_right_camera_sensor/image_raw	sensor_msgs/msg/Image
/wamv/sensors/cameras/front_right_camera_sensor/optical/camera_info	sensor_msgs/msg/CameraInfo
/wamv/sensors/cameras/front_right_camera_sensor/optical/image_raw	sensor_msgs/msg/Image
/wamv/sensors/cameras/middle_right_camera_sensor/camera_info	sensor_msgs/msg/CameraInfo
/wamv/sensors/cameras/middle_right_camera_sensor/image_raw	sensor_msgs/msg/Image
/wamv/sensors/cameras/middle_right_camera_sensor/optical/camera_info	sensor_msgs/msg/CameraInfo
/wamv/sensors/cameras/middle_right_camera_sensor/optical/image_raw	sensor_msgs/msg/Image
/wamv/sensors/gps/gps_fix	sensor_msgs/msg/NavSatFix
/wamv/sensors/imu/imu_data	sensor_msgs/msg/Imu
/wamv/sensors/lidars/lidar_wamv_sensor/points	sensor_msgs/msg/PointCloud2
/wamv/sensors/lidars/lidar_wamv_sensor/scan	sensor_msgs/msg/LaserScan
/wamv/shooters/ball_shooter/fire	std_msgs/msg/Bool
/wamv/thrusters/left/pos	std_msgs/msg/Float64

Message Publisher

topic	type	rate	expression
<input checked="" type="checkbox"/> /wamv/thrusters/left/thrust	Float64	5.00	
<input checked="" type="checkbox"/> data	double	20	
<input checked="" type="checkbox"/> /wamv/thrusters/right/thrust	Float64	15.00	
<input checked="" type="checkbox"/> data	double	80	

# Perception

- Use LiDAR to locate the target markers, employing DBSCAN to cluster the point cloud data.
- Fuse the camera and LiDAR outputs in the format, `<Object_id, position_LiDAR_frame>`.
- Convert the marker positions from the LiDAR coordinate frame to the geographic coordinate system and publish the final results.



# More Challenges

## Acoustic Tracking



## Scan and Dock and Deliver





The background of the slide is a 3D simulation of a robotic boat on a body of water. The boat is a catamaran-style vessel with two long, narrow hulls. It has a central mast with a camera or sensor mounted on it. There are some boxes and equipment on the deck. In the background, there is a shoreline with trees and a building. A large, semi-transparent purple speech bubble is overlaid on the image, containing the text "Thank You!".

Thank You!