MarineSIM : Robot Simulation for Marine Environments

P.G.C.Namal Senarathne*, Wijerupage Sardha Wijesoma*, Kwang Wee Lee†, Bharath Kalyan*, Moratuwage M.D.P*, Nicholas M. Patrikalakis‡, Franz S. Hover‡
*School of Electrical and Electronic Engineering,
Nanyang Technological University, Singapore
†DSO National Laboratories, Singapore
‡Department of Mechanical Engineering,
Massachusetts Institute of Technology, Cambridge, MA

***Abstract*—Development of robust navigation algorithms for marine robotics is a challenge faced by many marine robotists. This paper presents MarineSIM, a marine robot simulation platform which provides an infrastructure to easily test and debug navigation algorithms before they are being deployed in the actual environment. The general architecture of the simulation platform is introduced together with the marine robot models and sensor models used. A detailed example is provided on how to use this platform to simulate a collaborative mapping of a marine environment using both above waterline and below waterline sensors.**

I. Introduction

Autonomy in marine robotics is a very desirable feature. Consider a marine application of robotics where a fleet of surface water and underwater vehicles are used to map and gather data in a dynamic coastal environment such as a harbor. These vehicles should be able to avoid obstacles, follow correct navigation paths and coordinate among each other to successfully complete a desired goal. This requires accurate perception of the coastal environment using sensors and fusing these sensor data to generate information that can be used by the on board computer to make intelligent navigation decisions in real-time. Therefore correct and efficient processing of real-time data is one of the most important tasks in marine robotics. Hence the programs used to handle the vast amount of data gathered from sensors need to be tested extensively before deployment. However there is a major problem restricting the researchers from carrying out such testing. That is the number of field experiments conducted in a particular period of time.

Field experiments in marine robotics are hindered by the huge financial costs incurred in actual deployment and testing. Planning and management burden of a large collection of resources, adhering to safety regulations at coastal environments [1] and experiment time spanning several days or weeks are also major issues. These factors limit the number of field experiments that can be carried out in a particular period of time. Therefore simulation of navigation of marine robots in coastal and sea environmental conditions together with acceptable sensor data is a prime requirement for marine robotics community in the development of effective algorithms.

Taking above into consideration, we have developed a simulation platform which enables us to model and simulate different sensors, robots and application scenarios in the marine environment. We call it MarineSIM and it is an integration of three main open source software components released by several institutions and several robots and sensors modeled in-house. Fig. 2 is a simplified block diagram which depicts how these three software components are integrated with each other to create an effective simulator platform.

Fig. 1. Conducting experiments in Singapore waters with SCOUT robot

II. Related Work

Mobility Open Architecture Simulation and Tools (MOAST) [2] and USARSim is used together as a combined framework for development and testing of autonomous vehicles [3] . MOAST follows the 4D/RCS [4] structure at its core. This structure is organized as echelons and MOAST provides the top four (Team, Vehicle, Autonomous Mobility, Primitive) echelons while the remaining Servo echelon is provided by USARSim. Testing of algorithms are done per echelon basis in most cases. So initially a baseline system is configured and a module in the desired echelon is tested. This layered approach allows isolated testing of algorithms with minimum variations in the other modules. Collaboration of multiple robots in MOAST happens only at the top most layer but with less granularity. But for a collaborative mult-robot setup, sharing of data at lower levels such as sharing map

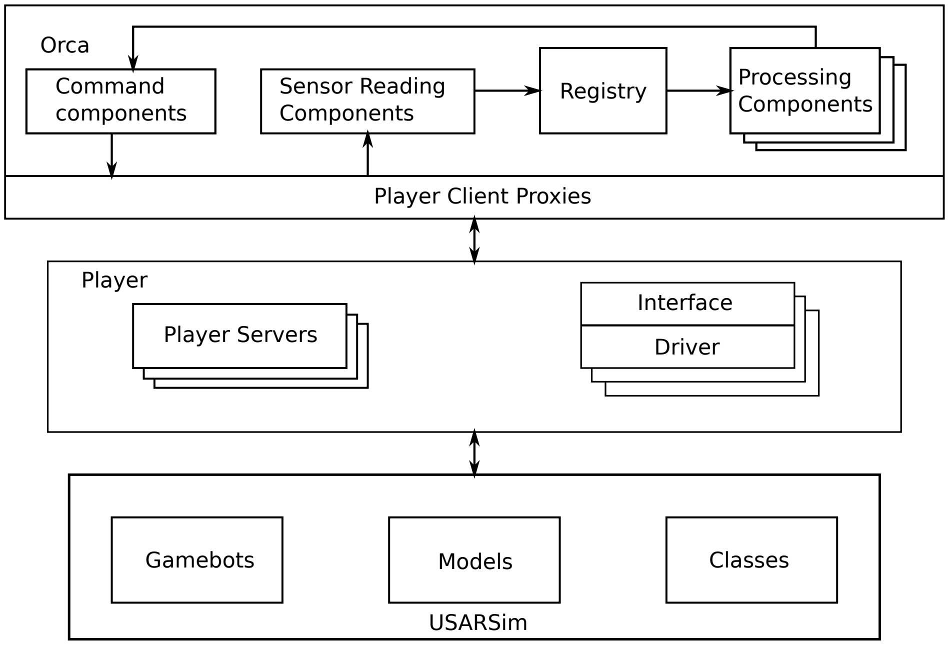

Fig. 2. Architecture of MarineSIM

information is vital. This forces the developer to explicitly program inter module communication in order to allow data sharing.

Multi Robot Control System (MrCS) [5] together with USARSim provides another development platform which allows researchers to focus on collaborative robot missions with possibly a human in the loop. It uses a middle-ware called Machinetta to allow all the robots and the human operator to collaboratively conduct a mission. MrCS is primarily used as a control tool for RoboCup virtual robot competition.

III. MarineSIM

A. Architecture of the simulator

USARSim [6] is a famous open source simulator for robotics research used by many researchers around the world. It manages robots, environments and interactions between the robots/sensors with the environment. *Player* [7] is an open source device server for robots. In this setup, player provides an easy way to access robots and sensors that are simulated and the data generated in USARSim. For each robot being simulated, an associated player server is created. The configuration file for the player server specifies the type of robot to be created and the set of devices that can be accessed from the created robot. A collection of player servers can be used to create a multi-robot environment in the simulator. *Orca* [8] is a framework to develop robot software in a component based fashion. That is, Orca provides the ability to compose a robot system using reusable software components rather than programming the entire system from scratch.

B. Advantages of the software stack

Several factors contributed to using USARSim as the foundation of the platform. The high accuracy in rendering and physical simulation provided by the game engine [9], Unreal Tournament 2, employed by USARSim is a major factor. For example, the movement of the Remus AUV is decided based on the values for propeller speed, rudder position and stern plane position. So even though rudders are positioned to give maximum turn, if the propeller speed is very low, the resultant turning rate is not high and vice-versa.

Another important feature is the extensibility of the simulator. The ability to model new robots, sensors and environments and the facility to seamlessly integrate them to the existing system is a very important property. It is possible to write controller applications which directly connect to USARSim over TCP/IP. But management of a huge collection of sensor data and implementing algorithms in a monolithic fashion makes these controllers very complex and very hard to debug and thus limit the use of USARSim as an independent simulation tool. The separation of parsing of sensor data from processing of sensor data is an elegant solution to this problem. Player and Orca are used to support data parsing and data processing respectively.

Player comprises of a collection of device drivers that can be used to parse data generated from the simulator. Player allows to organize these sensor data according to the robot ID. In a multi-robot marine environment this feature comes very handy as developers do not have to worry about managing sensor data according to robots at the code level, that task is taken care of by player servers.

Orca's ability to develop component based robot code is

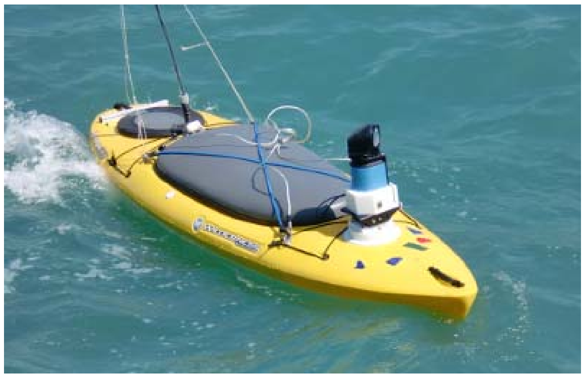
Fig. 3. SCOUT robot with 2D Laser

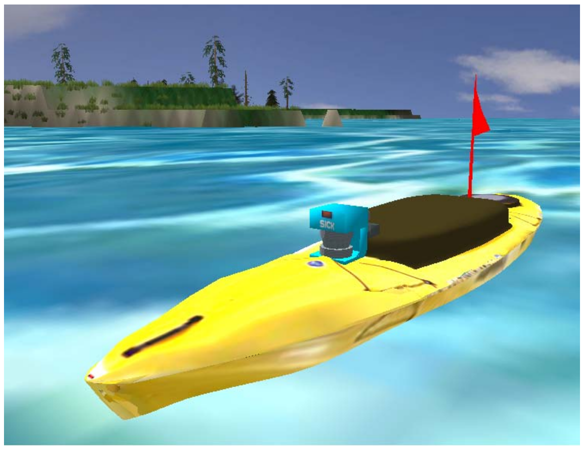
Fig. 4. Simulated SCOUT with 2D Laser

the main reason to integrate it into MarineSIM. Composing a robot system in contrast to programming is considered to be a valuable paradigm for researchers involved in robotics [10] and especially multi-robot systems. Consider a multi marine robot system. Generally all vehicles of a particular category (ASV or AUV) are not deployed with identical sensors. Most of the time they are equipped with complementary sensors. So when the programs are developed for the actual robots, all what needs to be done is to compose the robot controller using the components that match the available sensors for that particular robot. This makes it easier to generate controllers for mostly identical robots with possibly some differences in sensor mountings. Another important reason for using Orca is the ability to easily port these components to the real robot after testing in the simulator. This minimizes the amount of change that is required when porting code to the actual robot thus reducing the programming efforts and the operational failures of the controller. In addition, Orca provides an array of utility software such as data loggers and visualization tools which aides and expedites the development of robot code.

C. Models

As explained earlier, MarineSIM's simulations are done using the famous USARSim engine. It allows easy development of robot models and sensors using a scripting language called UnrealScript.

1) SCOUT: SCOUT [11] is a light weight Autonomous Surface Craft developed by MIT to support marine robotic experiments. Fig. 3 shows a SCOUT robot equipped with a 2D laser operating in Singapore waters during a field trial. Fig. 4 shows the simulated SCOUT in MarineSIM with a 2D laser. Current configuration of SCOUT in MarineSIM models and simulates sensors such as 2D Laser, Doppler Velocity Logger, Forward Looking Sonar and Inertial Navigation Sensor.

2) Remus: Remus is just one type of Autonomous Underwater Vehicle model used in MarineSIM. It is modeled after the Remus 100 AUV from Hydrioid Inc. The current configuration of Remus includes the DVL and INS sensor. Fig 5 illustrates the simulated model in USARSim.

3) Sensors: Two sensors used in marine environment are currently modeled in MarineSIM. Blue View MB1350 sonar

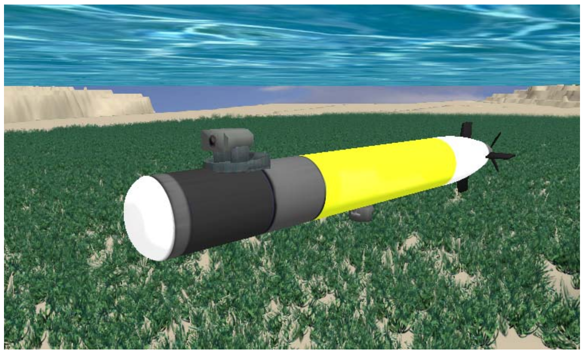
Fig. 5. Simulated REMUS AUV

is the main sensor that is used for underwater perception. Currently it is modeled as having a 45° Horizontal Field of View with a resolution of 256 data points per scan. Its range is set to 20 meters. These configurations can be easily altered in the simulator to support different models of sonars with different specifications. The Doppler Velocity Log (DVL) sensor is modeled to provide the instantaneous velocities of the vehicle.

IV. Experimental tests and results

Initial testing has established that MarineSIM can be used as a test bed for easy development of robust navigation

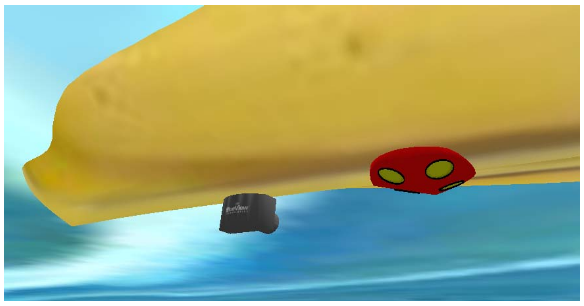
Fig. 6. Simulated Sonar (left) and DVL (right)

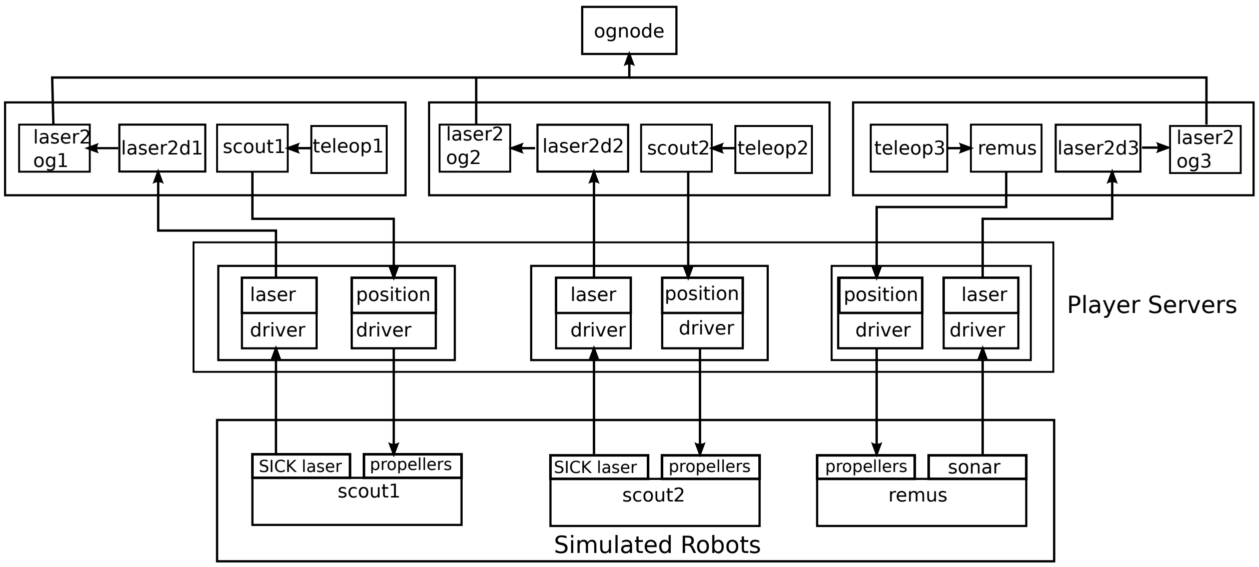

Fig. 7. Architectural arrangement for multi-robot mapping

algorithms. This section will focus on the utility of MarineSIM for simulations involving multi robot navigation.

A. Tele-operation

Testing of mapping and localization algorithms is accomplished by tele-operating the vehicles in the environment with the simulated sensors. The `teleop` component provided by Orca is directly used for this task.

B. Multi-Robot Mapping

The following test illustrates the flexibility of MarineSIM in supporting simulation of multi robot navigation scenarios. In this simulation, two Scout ASCs and one Remus AUV is used. The two ASCs are identical to each other and have a 2D Laser mounted on the front side. These two ASCs are employed for above water mapping. The AUV is equipped with a forward looking sonar and is used for below water mapping. Fig. 8 and 9 show the environment being mapped and the arrangement of robots. All data from both above waterline and below waterline sensors are fused together by a component in Orca to generate the global map. Fig. 7 illustrates the architectural arrangement of all the components used in this particular test scenario.

USARSim spawns three virtual robots corresponding to the two Scouts and the Remus. Then three player servers (running on three different ports) are created which correspond to the three robots simulated in USARSim. Each player server has a driver called `us_bot`. This driver is attached to an interface called `simulation`, and collectively the driver and the interface create the primary device which controls the corresponding robot in USARSim. For simplicity, both `us_bot` and `localized` drivers are aggregated into a single device as shown in fig. 7.

The laser device in the player server reads laser data from USARSim and presents the data using a standard interface. The same device is used to read sonar data from the AUV in USARSim. Even though operation of sonar and laser is different, the output data has similar structure, therefore for the sake of easy implementation, the same device is used.

Carrying forward the organized nature of sensor data, orca manages all the robots separately. Components are organized according to platforms. One platform corresponds to a single player server which in turn corresponds to a single simulated robot. This clear separation of data flow paths according to robots makes it very much easy to develop and test programs for multi robot navigation.

Orca components `scout` and `remus` are used to control the simulated ASCs and AUV respectively. These components provide a velocity control interface which is used by standard orca component `teleop` to navigate the vehicles. Standard orca components, `laser2d`, are used to read laser data presented from respective player servers and are fused using `orca2og` components. Finally these fused data is used by `ognode` to generate the occupancy grid map of the environment that is depicted in fig. 10. As it is shown in the fig. 7, there is a clear structure to the simulated scenario thus making it very easy to test, debug and extend the scenario by adding more sensors and more robots without the need for complicated programming.

V. Conclusion

Simulation of marine robotic environments with acceptable sensor data is important for development of robust controllers for AUV and ASV navigation. MarineSIM is a step towards realizing this goal. The architecture with an integrated simulation and development environment which allows researcher to isolate only the required data streams and allows fusion of multiple data streams for collaborative robot operation makes MarineSIM an ideal tool to be used in the initial phases of

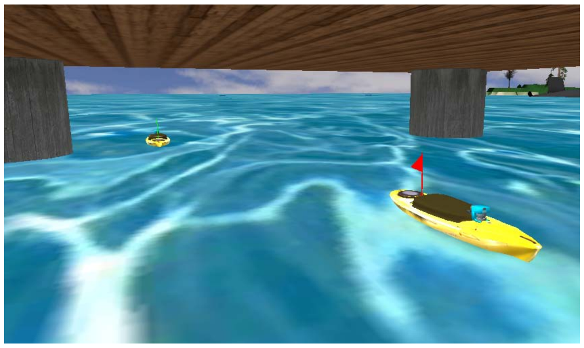

Fig. 8. Abovewater mapping

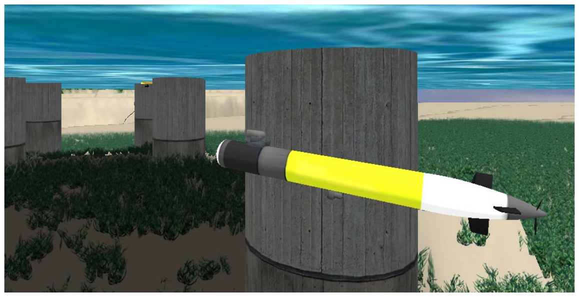

Fig. 9. Underwater mapping

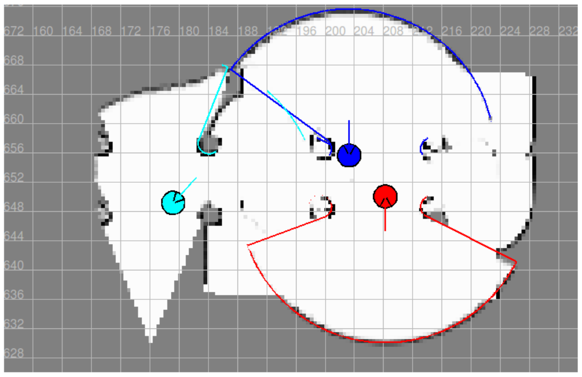

Fig. 10. Resultant map

algorithm development in cooperative marine robotics.

Acknowledgment

This work was supported by National Research Foundation (NRF), Singapore and Center for Environmental Sensing and Modeling (CENSAM) under the auspices of the Singapore-MIT Alliance for Research and Technology (SMART).

References

[1] Maritime and Ports Authority, Singapore, “Merchant Shipping Act,” 1996.
[2] “Mobility Open Architecture Simulation and Tools(MOAST) Project.” [Online]. Available: http://sourceforge.net/projects/moast/
[3] Stephen Balakirsky and Elena Messina, “MOAST and USARSim - A Combined Framework for the Development and Testing of Autonomous Systems,” in *Proceedings of the SPIE Defense and Security Symposium*, 2006.
[4] James S. Albus, “4-D/RCS reference model architecture for unmanned ground vehicles,” 2002.
[5] “Multi-robot Control System (MrCS) project,” 2010. [Online]. Available: http://athiri.cimds.ri.cmu.edu/foswiki/UsarSim/WebHome
[6] “USARSim (Unified System for Automation and Robot Simulation) Project.” [Online]. Available: http://sourceforge.net/projects/usarsim/
[7] “The Player Project.” [Online]. Available: http://playerstage.sourceforge.net/
[8] “Orca : Components for Robotics.” [Online]. Available: http://orca-robotics.sourceforge.net/
[9] S. Carpin, M. Lewis, J. Wang, S. Balakirsky and C. Scrapper, “USARSim: a robot simulator for research and education,” in *Proceedings of the IEEE Conference on Robotics and Automation*, 2007, pp. 1400–1405.
[10] A. Brooks, T. Kaupp, A. Makarenko, A. Orebck and S. Williams, “Towards Component Based Robotics,” in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2005, pp. 163–168.
[11] J. Curcio, J. Leonard and A. Patrikalakis, “SCOUT - A Low Cost Autonomous Surface Platform for Research in Cooperative Autonomy,” in *Proceedings of MTS/IEEE Oceans*, 2005.