VIRTUAL UNDERWATER LAB: EFFICIENT TOOL FOR SYSTEM INTEGRATION & UUV CONTROL DEVELOPMENT

E. Omerdic, J. Riordan, L. Molnar and D. Toal

Mobile & Marine Robotics Research Centre, ECE Department
University of Limerick, Limerick, Ireland
email: {edin.omerdic, james.riordan, levente.molnar,daniel.toal}@ul.ie

Abstract: This paper describes the main features of the Virtual Underwater Lab (VUL), the unique, generic solution for underwater development, developed by the Mobile & Marine Robotics Research Centre (MMRRC), UL. As part of the current research project, the MMRRC has acquired the state-of-the-art survey equipment, including multi-beam; inertial navigation system (INS); Doppler Velocity Log (DVL); depth/altitude sensors and digital precision altimeters. During research cruises in February and June 2005 with RV Celtic Explorer, weak points of the overall system integration were identified. The VUL is a mixed hardware/software tool, designed to overcome identified problems and make easier overall system integration. The VUL has an open architecture, providing a framework for researchers to develop, implement and test advanced control algorithms in a safe, simulated environment before actual tests are performed in real-world environment. The signal-level compatibility between simulated and real-world environment provides opportunity for rapid control prototyping and hardware-in-the-loop development techniques to be used in system design.

Keywords: Simulators, Hardware-in-the-loop, Rapid control prototyping, Real-time visualisation, Virtual reality.

1. INTRODUCTION

This paper describes the main features of the Virtual Underwater Lab (VUL), the unique solution for underwater development developed by the Mobile & Marine Robotics Research Centre (MMRRC), UL. As part of the current research project (HEA PRTLI III "Deep Ocean Habitat Mapping Using an ROV"), the MMRRC has acquired the state-of-the-art survey equipment, including a high-resolution multi-beam sonar SeaBat 7125; an Ixsea PHINS inertial navigation system (INS); an RDI Doppler Velocity Log (DVL); a MicroBath depth sensor, six Tritech digital precision altimeters for obstacle avoidance, National Instruments Compact RIO and Compact Vision systems. During research cruises in February and June 2005 with RV Celtic Explorer, weak points of the overall system integration were identified. The VUL is a mixed hardware/software tool, designed to overcome identified problems, to make easier overall system integration and to provide a framework for researchers to develop, implement and test advanced control algorithms in simulated virtual environment, under conditions very similar to the real-world environment. The paper expands upon the work previously reported by the authors (Omerdic and Roberts, 2006; Molnar, *et al*., 2005). Recent advances in computer technology made it possible to perform data acquisition, processing, analysis and visualisation with high speeds, previously considered infeasible. One such application is real-time, human-computer interface described in Martin, *et al*. (2006). Excellent review of current research activities in the field of development of hardware-in-the-loop simulators is given in Ridao, *et al*. (2006).

Motivation for the VUL development is described in the second section. The third section gives the overview of the VUL. Details about implementation are given in the fourth section. Finally, the fifth section summarizes the concluding remarks.

2. MOTIVATION

The MMRRC is developing the VUL as integrated solution to fulfil research objectives and needs under implementation constraints, as described in the following:

Need to use rapid control prototyping (RCP) and hardware-in-the-loop for system development: The exploration and exploitation of the deep oceans is a challenging, expensive and time-consuming activity. Due to the complexity of the underwater environment, the vehicle control system must avoid obstacles and compensate for various external disturbances acting on the vehicle (sea currents, drag effects of umbilical for ROVs, etc.) in the presence of uncertainties in determination of position and orientation (navigation errors). The physical shape and actuator configuration of the vehicle (number, position and orientation of thrusters & control surfaces) impose constraints that limit the freedom of control actions. All these factors add significant challenges to the task of maintaining the vehicle on the desired path and performing the survey mission. If the control system is not designed in an optimal way, tracking errors lead to unnecessary extension of survey mission time, increasing the survey mission costs. In addition, a non-optimal control allocation of actuators leads to inefficient usage of available energy resources. For an AUV, this can lead to undesired cancellation of the survey mission, due to lack of energy (discharged battery). Intelligent control systems with high degrees of autonomy should perform well under significant uncertainties for extended periods of time, and they must be able to compensate for certain system failures without external intervention. Dynamic behaviour of the vehicle depends upon the configuration of the onboard sensors and equipment. In order to achieve the best performance, low-level controllers need to be retuned before each mission to adapt to particular vessel configuration. *In order to develop, implement and test advanced control algorithms with auto-tuning and fault-tolerant capabilities under realistic conditions, it is desirable to build environment, where rapid control prototyping and hardware-in-the-loop techniques can be used.*

Need to test connection and system integration before the mission execution: Typical seabed survey mission requires integration of the existing ship equipment (GPS, USBL) with ROV onboard components (Multi-beam, INS, DVL, Depth sensor, Sound velocity probe, Vision system, etc.). Each component has its own mechanical and electrical interface (mechanical installation & alignment, power requirements, communication protocols, etc.). Most sensors for underwater measurements still use ordinary serial protocols for communication, which makes wiring of these components more complex. In addition, each component has its own configuration software. In order to obtain the best quality of navigation data, it is necessary to reconfigure some components for the best performance in real-time, depending on the stage of the mission. Because of all these reasons, integration of the overall system is not trivial task and requires significant technical expertise. It is also very expensive to work with all this equipment in real conditions, since ship time is very expensive (about €15000 per day). *Hence, it is desirable to develop simulated environment, where all connections and functionality can be checked in the lab and many integration problems can be detected and resolved in advance, before the actual mission is performed.*

Need to develop a set of aiding tools for ROV pilot: Modern ROVs perform sampling, data acquisition and high-resolution acoustic and video survey in deep oceans. Expensive equipment is deployed very close to the seabed and the ROV pilot has a huge amount of responsibility to control the vehicle and prevent any damage or loss of equipment. The success of missions is closely related to individual skills and previous experience of ROV pilots. However, it would be of a great importance and potential commercial applications to develop a set of aiding tools (control algorithms) which will enable that an ROV pilot with moderate skills perform complex tasks easy, with excellent performance and final results. His task will be to bring the vehicle to the initial position, to activate corresponding aiding tool (algorithm) and to monitor the realisation of the task. *In order to develop a set of aiding tools, it is desirable to have simulated environment, where different control algorithms can be implemented, tested and refined, before real-world tests.*

Need to provide 3D real-time visualisation: In underwater environment, ROV pilot's field of view is limited exclusively to onboard cameras. Great assistance for steering would be the side view of the vehicle and its underwater environment. This can be possible, if virtual reality (VR) underwater scene with 3D models of the vehicle, ship and seabed is prepared in advance and real-time signals from sensors (position and orientation of the ROV and ship) are forwarded to the VR scene in real-time. Different view points in the VR scene can be rendered in different displays, such that the same scene can be viewed from the different angles at the same time. In this way, it could be possible to visualise navigation data in real time, which can make control of the vehicle easier and safer. *In order to visualise navigation data in real time, it is desirable to have simulated environment with efficient virtual reality visualisation capabilities.*

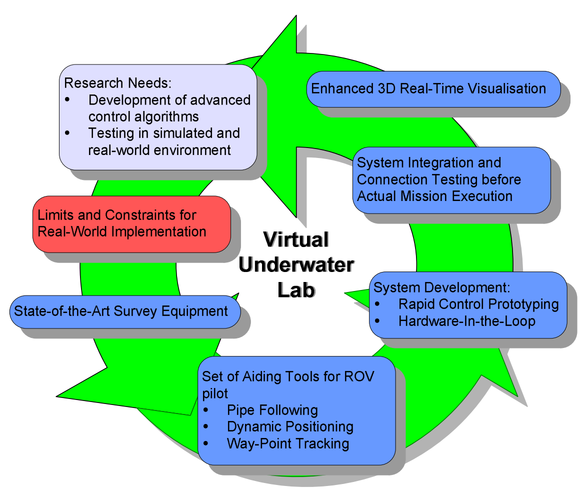

Fig. 1. The Virtual Underwater Lab: tool for system integration & control development.

3. SYSTEM OVERVIEW

In order to fulfil all these needs in an efficient way, the MMRRC is developing the Virtual Underwater Lab (see Fig. 1.).

The main features of the VUL are:

- Signal-level compatibility between simulated and real-world environment,
- 3D real-time visualisation of navigation data,
- Advanced, flexible, fault-tolerant control system with auto-tuning capabilities,
- Open architecture for rapid control prototyping and hardware-in-the-loop development,
- Set of aiding tools for ROV pilot.

Interconnection between the VUL components and existing ship and ROV equipment is shown in Fig. 2. The role, location and input/output interface for main hardware components in real-world implementation is given in Table 1. In order to use the VUL in simulated environment, real-world components from the ship (GPS1, GAPS) and the ROV (PHINS, external sensors, power lines and leak detectors) are replaced with hardware/software simulators, as shown in Fig. 3 and Table 2.

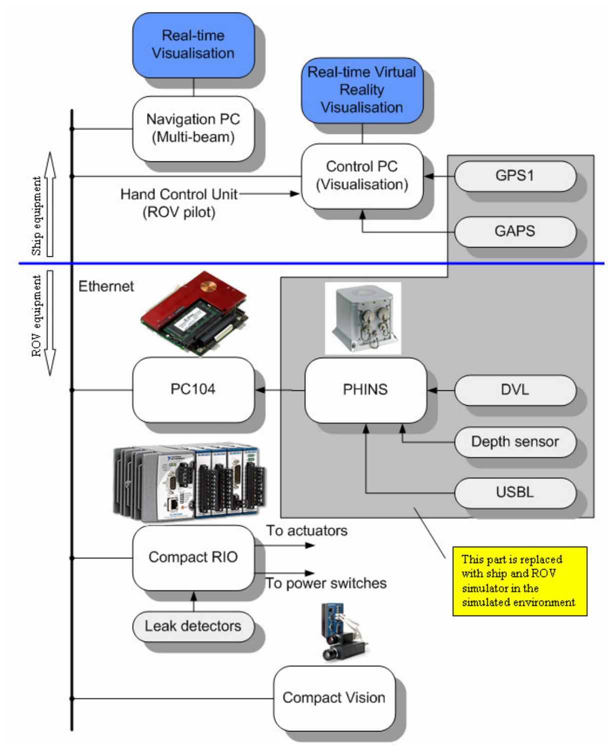

Fig. 2. Real-world VUL implementation.

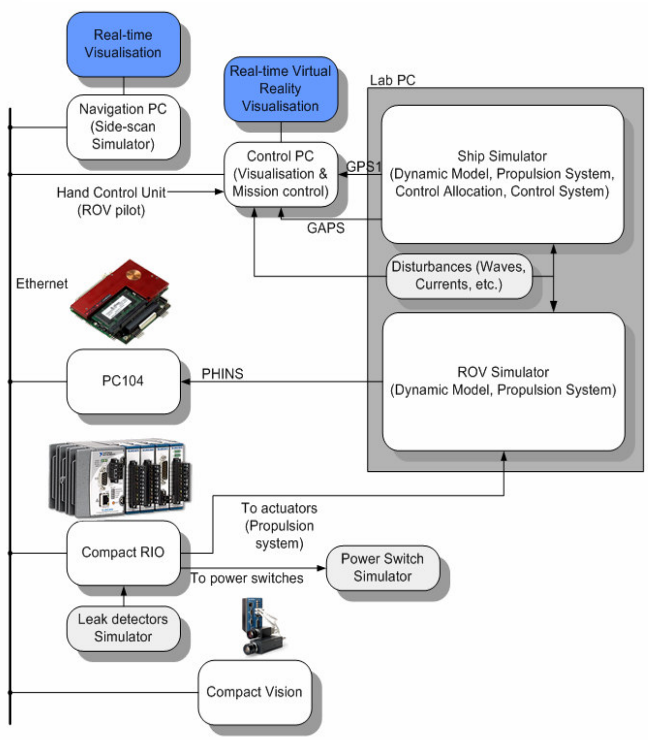

Fig. 3. VUL implementation in simulated environment.

Table 1 Components in real-world environment

Component	Location	Role	Input(s)	Output(s)
GPS1, GAPS	Ship	Absolute position and orientation of the ship	Signals from GPS satellites, GAPS internal INS	PHINS STANDARD protocol
Navigation PC	Ship	Acquisition, analysis and visualisation of the multi-beam data	Multi-beam data (bathymetry)	Real-time visualisation of multi-beam bathymetry
Control PC	Ship	3D real-time visualisation; Mission control & Management (task planning); Read data from HCU; Remotely control of power lines for other components	Absolute position and orientation of ship (PHINS STANDARD) and ROV (PHINS STANDARD shared from PC104); HCU data	Real-time 3D visualisation of the ship & ROV in virtual reality environment
PHINS	ROV	Absolute position and orientation of the ROV (Kalman filter used internally to integrate INS and external sensors (Depth, DVL, USBL))	External sensors (Depth, DVL, USBL)	Absolute position and orientation of the ROV (PHINS STANDARD); Linear velocities and rotation rates (PHINS CONTROL)
PC104	ROV	Read, process & share PHINS data; Fusion of different tasks; Send final control input to Compact RIO	Absolute position and orientation of the ROV (PHINS STANDARD); Linear velocities and rotation rates (PHINS CONTROL)	Absolute position and orientation of the ROV (shared variable), Control cluster (shared variable); Final control input to Compact RIO
Compact RIO	ROV	Control allocation; Thruster fault detection and accommodation; Control power lines of other components; Data analysis from leak detectors; Truly parallel control of actuators	Final control input from PC104; Leak detectors	Control signals to actuators; digital outputs to control power lines of other components
Compact Vision	ROV	Application of machine vision in the control (Station keeping, Feature tracking, Video mosaicking)	Video signals from onboard cameras	Features of surrounding objects

Table 2 Components in simulated environment

Component	Location	Role	Input(s)	Output(s)
Ship Simulator (Software)	Lab PC	Simulate dynamics of the ship (Absolute position and orientation); control system includes way-point tracking and dynamic positioning features	Wave, wind & current disturbances	PHINS STANDARD protocol
Navigation PC	Lab	Real-time side-scan/multi-beam simulator	Absolute position and orientation of the ROV from the ROV simulator (PHINS STANDARD)	Real-time visualisation of simulated side-scan/multi-beam image
Control PC	Lab	3D real-time visualisation; Mission control & Management (task planning); Read data from HCU; Remotely control of simulated power lines for other components	Absolute position and orientation of ship (PHINS STANDARD from ship simulator) and ROV (PHINS STANDARD – shared variable originated from PC104); HCU data	Real-time 3D visualisation of the ship & ROV in virtual reality environment
ROV Simulator (Software)	Lab PC	Simulate 6 DOF dynamics of the ROV (Absolute position and orientation in the earth-fixed frame; Linear and angular velocities in the body-fixed frame)	Wave & current disturbances; Actuator settings	Absolute position and orientation of the ROV (PHINS STANDARD); Linear velocities and rotation rates (PHINS CONTROL)
PC104	Lab	Read, process & share PHINS data; Fusion of different tasks; Send final control input to Compact RIO	Absolute position and orientation of the ROV (PHINS STANDARD); Linear and angular velocities (PHINS CONTROL)	Absolute position and orientation of the ROV (shared variable), Control cluster (shared variable); Final control input to Compact RIO
Compact RIO	Lab	Control allocation; Thruster fault detection and accommodation; Control simulated power lines of other components; Data analysis from simulated leak detectors; Truly parallel control of actuators	Final control input from PC104; simulated signals from leak detectors	Control signals to actuators (part of ROV simulator); digital outputs to control simulated power lines of other components
Compact Vision	Lab	Application of machine vision in the control (Station keeping, Feature tracking, Video mosaicking)	Video signals from simulated onboard cameras in VR scene	Features of surrounding objects

The Navigation PC is running a real-time side-scan/multi-beam sonar simulator, developed by J. Riordan from the MMRRC, UL. The serial input to the sonar simulator is PHINS STANDARD (absolute position and orientation of the ROV in the Earth-fixed frame obtained from the ROV simulator). Alternatively, the same data can be sent through Ethernet using the UDP protocol. More details about integration with the sonar simulator can be found in Riordan, *et al*. (2006).

It should be emphasised that all simulators are synchronised with real time. Full 6 DOF vessel dynamic models are implemented, including thruster DC-motor dynamics with nonlinearities, such as saturation, slew-rate limiter, friction, nonlinear propeller load etc. Different components of ship and ROV simulators are simulated as parallel loops executed with different speeds, depending on dynamics of components. For example, the Propulsion System component for ROV simulator has simulation time 1 ms synchronised with real-time, so it is possible to monitor current consumption of thruster motors in real-time. However, the rigid body dynamic loop is executed with simulation time 200 ms, compatible with PHINS output rate.

The inputs to and outputs from virtual instruments (simulators) are compatible with corresponding real-world instruments on the signal level. This means, for example, that the output PHINS STANDARD of the ROV simulator is sent through the serial ports in the same RS232 format as output of real PHINS (Identical message formats including checksums; Port settings are: Baud rate 57600, Parity: Odd, Stop bits: 2). In this way, all communication delays and latencies are present during the control design stage, which provide a framework to design robust control system in realistic environment.

The ROV control system utilizes a hybrid control approach, which combines the advantages of a top-down hierarchical AI approach and a bottom-up behaviour-based approach. Currently, two high-level task executors have been implemented: Way-Point Tracking and Obstacle Avoidance (Molnar, *et al*., 2005, see Fig. 4).

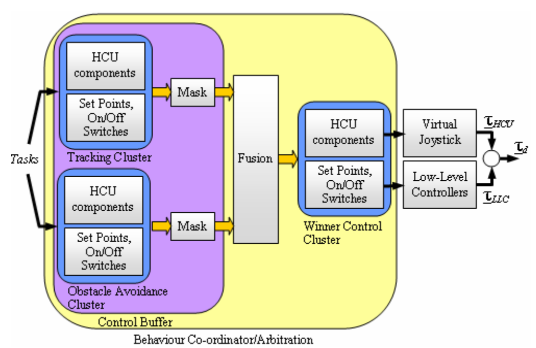

Fig. 4. Behaviour Co-ordinator/Arbitration component.

Each of these task executors is competing to take control of actuators. The control buffer concept has been developed to provide transparency and easy fusion of different task executor control demands. Each task executor has its own control cluster and mask inside the control buffer. The control cluster consists of Hand Control Unit (HCU) components (to simulate a virtual joystick) and a set of settings for low-level controllers (set points and on/off switches to activate/deactivate individual controllers).

Each control cluster is masked with a corresponding mask, depending on the state of the mission and navigation data (interaction with the real world). A mask is built from weights and logic gates, which are bundled into the same structure as the control cluster. Masking is performed by multiplying (ANDing) corresponding fields in the control cluster and mask. The priority level of the task executor is determined by the mask content. In this way, it is possible to control the degree of co-operation and competition between different task executors. After masking, control clusters are bundled into the Winner Control Cluster, which has an exclusive actuator control.

4. IMPLEMENTATION

All software (except sonar simulator) is implemented in NI LabVIEW using State Diagram Toolkit, Simulation Module and FPGA Module. Data (outputs of individual components) are bundled into clusters and transmitted using network-published shared variables, based on the NI Publish-Subscribe Protocol (NI-PSP). The NI-PSP protocol uses less network bandwidth and is more efficient than TCP/IP for a given requirements of the NI-PSP protocol. Unlike the UDP protocol, however, the NI-PSP protocol guarantees delivery by implementing an additional layer of functionality on top of the raw UDP protocol.
A set of LabVIEW toolkits has been developed, including VIs for quaternions, kinematics transformations, geographic data transformations and special purpose signal conditioning. A library of Express VIs has been developed using these toolkits, including Control Allocation, Joystick Conditioning, PHINS Simulator & Conditioning, and Virtual Reality Interface Express VIs.

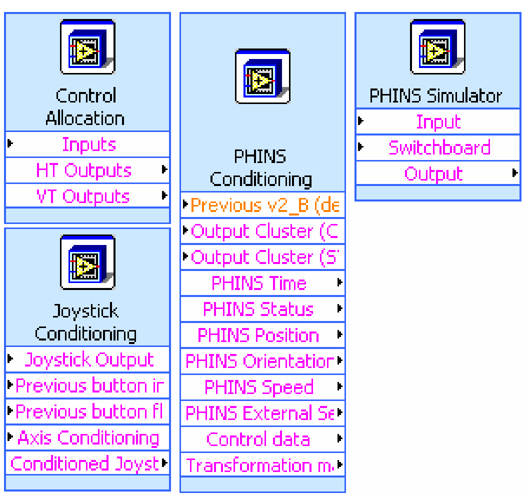

Fig. 5. Selection of Express VIs from a library.

4.1 Control allocation

A hybrid control allocation approach (Omerdic and Roberts, 2006) is implemented inside the Control Allocation Express VI. In fault-free case, optimal control allocation is guaranteed for all possible command inputs, since the hybrid approach for control allocation finds the exact solution on the entire attainable command set. This solution is optimal in the l_2 sense, i.e. it minimises a control energy cost function, the most suitable criteria for underwater applications. In faulty situations, the fault diagnosis part of the system immediately detects and isolates any fault in a thruster using fault detection units and delivers knowledge about faults in the form of a fault indicator vector. The fault accommodation part of the system uses this vector to accommodate fault and eventually switch off the faulty thruster. At the same time, control reallocation is performed by redistribution of control energy among remaining operable thrusters, such that the mission can be continued with a minimal loss of control performance. All inputs to the Control Allocation Express VI are normalised to standard intervals. Depending on user-defined settings inside the dialog box, the outputs (actuator settings) can be compensated for non-symmetrical propeller T-curves and adapted (scaled and/or rounded, if necessary) to meet requirements of thruster input interfaces. Degree of usage of each thruster is controlled by sliders HT (VT) Saturation Bounds. The position of these sliders is determined by ROV pilot or fault accommodation system, depending on state of thrusters (healthy, partial fault or total fault).
The control allocation problem is decomposed into motions in horizontal and vertical planes. Current version of the Control Allocation Express VI covers most common thruster configurations from one up to four thrusters. The user can choose combination that matches the particular thruster configurations of his vehicle.

4.2 Visualisation of thruster velocity saturation bounds

Visualisation of thruster velocity saturation bounds, i.e. attainable command set, implemented as part of the fault diagnosis and accommodation system (FDAS), provides insight into the constraints imposed by particular thruster configuration. During missions, the ROV pilot / control law generates command inputs, which stretch out over the command space in different directions. Thruster velocity saturation occurs in some directions easier than in other. The position of saturation bounds inside the command space is determined by thruster configuration. In order to avoid thruster velocity saturation, command inputs should not stretch outside the saturation bounds. Any fault in a thruster causes a change in the shape of the attainable command set. The FDAS provides a dynamic update of saturation bounds such that the ROV pilot / main controller are informed with the effects of thruster fault accommodation, incorporated in the new shape of the attainable command set. Using different indicators and visualisation tools, such as Thruster Velocity Saturation Indicator and Virtual Control Space Visualisation Tool, the FDAS informs the ROV pilot / main controller about position of actual command inputs relative to the actual attainable command set. Using this information, even an inexperienced ROV pilot is able to detect the situation when thruster velocity saturation occurs and to correct the command inputs such that it becomes attainable (feasible). In this way, the ROV pilot / main controller can easy adapt to newly created changes and continue a mission in presence of thruster fault(s).

4.3 Joystick conditioning

A joystick is used to simulate Hand Control Unit (HCU), used by ROV pilots to control vessels in real-world applications. The Joystick Conditioning Express VI performs processing of raw joystick output data using scaling, dead zone, saturation etc. in order to obtain a device functionally equivalent to the HCU.

4.4 PHINS conditioning & PHINS simulator

The PHINS Conditioning Express VI is used to extract navigation and control data from PHINS STANDARD and PHINS CONTROL messages.
The PHINS Simulator Express VI is used to generate PHINS-compatible message (PHINS STANDARD format), which is sent through COM port.

4.5 ROV and ship models

Two ROV models with different thruster configurations are implemented in the current version of the VUL: FALCON (Seaeye Marine Ltd, Fareham, UK) and TETHRA (MMRRC, UL, Ireland). A model of RV Celtic Explorer (Marine Institute, Galway, Ireland) is implemented in the current version of the VUL (Fig. 6.).

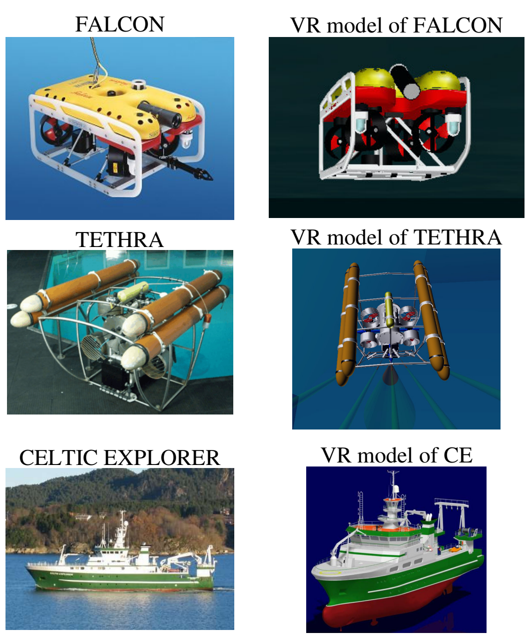

Fig. 6. ROV and ship models currently implemented.

4.5 Virtual reality interface

Virtual Reality Interface has been built using COM object technology. Virtual underwater scene is built in three steps, using the following wizards (Fig. 7.):

- Terrain Builder Wizard
- ROV Configuration Wizard
- Underwater Structure Configuration Wizard

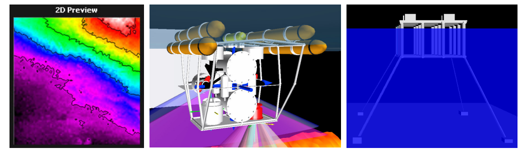

Fig. 7. Wizards used to build VR scene.

4.6 3D real-time visualisation

Navigation data from real-world/simulated environment are visualised using VR interface in real-time. The same VR scene can be viewed from different view points in separate displays at the same time (Fig. 8.).

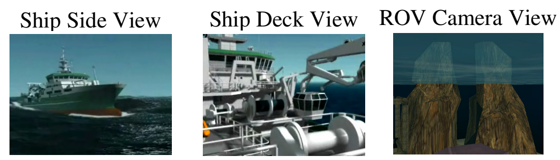

Fig. 8. VR scene seen from different view points.

5. CONCLUSIONS

This paper described current state in development of the VUL, generic hardware/software tool for integration of survey equipment with existing ROV and ship resources. The VUL has open architecture, providing a framework for researchers to develop, implement and test advanced control algorithms in safe simulated environment before actual tests are performed in real-world environment.

REFERENCES

Martin, S.C., L.L. Whitcomb, R. Arsenault, M. Plumlee and C. Ware (2006). Advances in real-time spatio-temporal 3D visalisation for underwater robotic exploration. Chapter 14 in: *Advances in Unmanned Marine Vehicles*, IEE Control Engineering Series.

Molnar, L., E. Omerdic and D. Toal (2005). Design of advanced AUV for deployment close to seabed and other hazards. *Ocean 2005 Europe IEEE Conference*, June 2005, Brest, France.

Omerdic, E. and G.N. Roberts (2006). Control Allocation for Over-Actuated Thruster-Propelled Underwater Vehicles. Chapter 5 in: *Advances in Unmanned Marine Vehicles*, IEE Control Engineering Series.

Ridao, P., E. Batlle, D. Ribas and M. Carreras (2004). NEPTUNE: A HIL Simulator for Multiple UUVs. *Oceans'04 MTS/IEEE*.

Riordan, J., E. Omerdic, L. Molnar and D. Toal (2006). Real-Time Sonar Simulator Integrated with Vehicle Control and Navigation for Ocean-Floor Mapping Applications in Survey-Scale Environments. *7th IFAC Conference on Manoeuvring and Control of Marine Craft (MCMC 2006)*, Lisbon, Portugal.