Proc. of IEEE Conf. on Autonomous Underwater
Vehicle Technology '96, June 3-6, Monterey, CA

AUV Test using Real/Virtual Synthetic World

Yoji Kuroda*, Koji Aramaki**, *and* Tamaki Ura**

*Department of Mechanical Engineering, School of Science and Technology,
Meiji University, 1-1-1 Higashi-mita, Tama-ku, Kawasaki 214 Japan

**Institute of Industrial Science, University of Tokyo,
7-22-1, Roppongi, Minato-ku, Tokyo 106 Japan

E-mail: ykuroda@isc.meiji.ac.jp

Abstract - This paper proposes a method for the research of underwater robots using technique of testing real/ virtual synthetic world. The world which is virtually created using the sensory information of both the real and a virtual world is called "synthetic world." In this world, the robots behave as if they were swimming in the underwater-world even in the case they are deployed in a simple shaped testing-pool. By testing in the synthetic world, the efficiency of underwater systems' development is improved because the software can be developed directly on the embedded computer system and the hardware and software cross-checking can be easily conducted. This paper shows the detailed descriptions and unique characteristics of the proposed testing method and the practical example for underwater robots.

I. INTRODUCTION

When we are going to develop an underwater robot, designing and implementing the robot management software which allows to carry out the mission are one of the most difficult part. The one reason is, as given by D. Brutzman (1992), inaccessibility of the robot. Unlike most other mobile robots, AUVs must maneuver unattended and uncontrolled in a remote and unforgiving environment. Inaccessibility complicates evaluation, diagnosis and correction of AUV system faults. And another reason is inexperience that AUVs (and also researchers) don't work enough around in actual undersea environment. Inexperience leads to unclarity where we stand or where we should go for the system designing. For example, could AUVs work correctly in the penetrated and overhanging rocks? Are the AUVs equipped with appropriate, enough amount of sensors? Does the data processing or obstacle recognition work correctly? To answer these questions, significant experience in the actual environment should be required. This tendency might be remarkable for the robot with higher intelligence which must take pictures or do other stationary jobs at close of the sea bed. Though it is time-consuming that examining whether the robot behaves correctly in such a complex environment, it should be done prior to launching for the official mission.

In this paper, we propose a testing method using "synthetic world" as an answer to experience undersea environment without risk of losing robots. The synthetic world is an ideal world which is the combination of both the actually measured and the virtually created sensory data, and can realize the hardware-in-the-loop testing in the realistic undersea environment.

The synthetic world is created by using virtual underwater-world simulator "MVS (Multi-Vehicle Simulator)." The MVS was designed as an integrated developing environment which can deal with the various developing stages from the very beginning of considering mission strategies of a robot up to the end prior to launching it to the ocean.

An experiment that an underwater robot navigates in a synthetic world is demonstrated in an actual testing pool with four virtual obstacles. The synthesis is carried out using the ranging sonar with which the actual and virtual underwater robots is equipped. The actual robot navigates in the synthetic world with avoiding both actual pool-walls and virtual obstacles.

II. SYNTHETIC WORLD

Since a mobile robot moves around in its workspace by referring environmental sensory data such as visual image, range map, etc., appropriate virtually created data can be regarded as those of the real world by swapping certain actual sensory data for corresponded virtually created data. When some parts of actual data are swapped for the virtual data, the world called "synthetic world" is created. Using the way of synthetic world, virtual agents such as robots, obstacles, seabeds, etc., can be deployed in the real world as shown in Fig. 1. The actual robot in the synthetic world can recognize and communicate to both actual and virtual agents without distinction. In the research field of "Augmented Reality," superimposed visual image is often used as a synthetic world for visualizing hardly-seen objects (e.g., inside view of patient's abdomen, see M. Bajura 1992).

Therefore, the proposed synthetic world can be regarded as an Augmented Reality for the robots.

A. Advantages of using a synthetic world

With this synthetic world, quality and efficiency of underwater systems' development can be improved based on the following special features;

1) It's not a simulation but an experiment using the actual robot including dynamics of the actual robot, actuators fitted on it, and some of sensors. The hardware/software cross-checking can be easily conducted.
2) It realizes an undersea environment in a simple testing tank. This contributes to making robust robots without the risk of loosing the under-developing robots in the real environment. Since the settings of the virtual world is re-configurable according to the objective of the experiments and the condition of the actual world, the quick and minute examining can be realized.
3) It is a part of the integrated developing steps shown in Fig. 2. The research work carried out in the seamless developing environment is improved in efficiency.
4) It might be realize the big scale experiments which include many types of actual and virtual robots, obstacles, seabed and people involved, and are connected to the international community through world wide networks.

B. Creating a synthetic world

In this section, an example of the creating scheme of a synthetic world is shown using the ranging-sonar data as a simple example. The following sensory data creation is carried out as shown in Fig. 3.

1) Preparation: At first, a virtual world which is customized for the actual world and the robot's sensors is prepared by using the virtual world simulator. The customization is considered for many ways, i.e., adding virtual obstacles, adding virtual robots, changing the shape or position of pool-walls, etc. The way of adding a virtual obstacle is shown in the following example.
2) Measure in real: An actual robot measures a set of range data and robot's current position and orientation in the real world coordinate.
3) Measure in virtual: After getting the actual robot's position and orientation, the virtual world simulator deploys the virtual robot, "ghost", on the corresponded position in the virtual world. The simulator calculates a set of virtual sensory-data of the ghost.
4) Synthesis: The proper data are selected from the actual or virtual sensory data for each sensor in every time step. The selection rule in this example is depended on a following equation;

$$R_{Si} = min\ (R_{Ri}, R_{Vi})$$

where, R_{Si} is a set of synthesized sensory data of the sensor i. R_{Ri} is the actual one measured in the real world. R_{Vi} is the virtual one calculated in the simulator.

Through these operations, the synthetic world is produced in real-time. From the robots, the produced synthetic world is seen like a motion picture with superimposed animated characters. In fact, the equation shown above is equivalent to the Z-buffering method for the three-dimensional computer graphics. Therefore, this synthesizing scheme is also applicable to the vision sensors of the robots.

III. VIRTUAL WORLD SIMULATOR: MVS

In the research field of underwater robotics, appropriate evaluation for the resulted behavior of the robot is difficult because the robot should be in underwater and it might be off the line during the operation. D. Brutzman (1992 a, b) developed an underwater-world simulator for the convenient to do research work of underwater robots in parallel with the research of an actual robot AUV-II. We have also been developing a virtual underwater-world simulator called the MVS (Multi-Vehicle Simulator) since 1992, especially for handling multiple vehicle simulation, and on-line works with actual robots (Y. Kuroda 1995, K. Aramaki 1995). The MVS was designed as shown in Fig. 2 as an integrated research and developing environment which can deal with the various developing stages from the very beginning of considering mission strategies of AUVs up to the end prior to launching them to the ocean. The basic concept of the MVS is "like real ocean, it should exist anytime on the computer networks, complex environment should be realized by the interaction with various entities including real/virtual robots, vessels, divers, obstacles, etc."

A. Features for providing realistic underwater environment

1) Multi-agent simulation: In order to produce realistic virtual environments, the MVS should be able to manage many agent processes that perform as robots, bottom stations, obstacles, human divers, seabed, etc., at a same time. The MVS is designed to have the distributed architecture as shown in Fig. 4, where the agents are connected each other through manager processes. The manager processes serve to the agents with data of the virtual world such as ranging sonar, ultrasonic command link, disturbance, collision, etc.

2) Multi-user accessibility: Since the coming cutting-edge underwater robotic systems should operate communicating with many other systems including bot-

tom-survey stations, charging stations, control centers through satellites, etc., the working style known as the collaboration would be a new, powerful method for developing robotic systems. On the advantage of multi-user-accessibility, work-group users can connect to the MVS through network and work together in the shared workspace.

3) Real-time performance & connectivity to the actual systems: The real-time simulation is useful for examining the availability of the developed software and hardware in the actual scene. The connectivity to the actual robotic systems is also brought by the real-time performance. This function makes the efficiency of underwater systems' development be improved because the software can be developed directly on the embedded computer system, and cross-checking of software and hardware can be easily conducted in virtually constructed world. The high connectivity to many actual/virtual agents can also provides a large scale simulation under worldwide network shown as Fig. 5. Since the researchers can access to any networked agents, it is realize that the MVS is utilized for tele-operation.

4) Multi-world & multi-CPU availability: Since the large scale simulation requires huge amount of computational and network resources, the simulation system should be carefully designed to reduce the resource consumption. The distributed architecture of the MVS provides multi-cpu configuration, the heavy processes can carried out on other platforms connected through the network. The multi-world creativity which is a unique feature of the MVS as shown in Fig. 6 allows to create some independently-existed virtual worlds to provide simulation environments to users without either interfere nor heavy network-resource consumption. Though current MVS function of the multi-world creativity supports the completely-independent configuration only, we have a plan to implement the loosely-connected worlds configuration. This feature provides zone clustering of simulating area, in order to realize the simulation of worldwide area by using the worldwide networked systems. In the systems, agents behave as the robots would be able to move over the networks, learn, and grow of their own performance.

5) Real-time rendered 3D visualization: The MVS allows connecting the graphic subsystems such that real-time-rendered three-dimensional visual imaging system which is an important function not only to examine the results of simulation intuitively but also to provide the visual image of the virtual environment to the actual vehicles which are equipped with vision sensors. Fig. 7 is an example of display images of a graphical user-interface program of the MVS and some control panels are shown.

6) Portable system: The programs used as the research environment with workgroups should have high portability. The MVS generally works on any type of processing units connected through any type of networks. In fact, the core programs of the MVS is currently running under SUN, SGI, SVR4 and Linux on PCs, and INMOS's Transputers. They are connected each other through TCP/IP or Transputer links. Though the graphical user-interfaces including 3D visualization function are currently running on SGI IRIS workstations, the text-base user interfaces are available on almost of all other systems.

B. The process coordination of the MVS

The process coordination of the MVS is shown in Fig. 8. The MVS processes consist of "managers" and "agents." The Universe manager is the unique, special manager to conduct all created worlds in the MVS. Each individual simulation is carried out in each world, so the following descriptions are focused on the inside of a world.

Each manager in a world is a process to provide the corresponded service to the agents, e.g., sonar and vision like environmental sensors, ultrasonic command-link like inter-agents communication, and collision and disturbance like physical phenomena, etc. Agents are the processes which behave as corresponded physical objects in a simulated world, e.g., autonomous robots, stations, obstacles, users, etc. In Fig. 8, a program executed as an agent is indicated as a "performer." Each agent is connected to the various managers in order to get required services, and communicates to other agents. In the MVS, the simulation is carried out as the results of communication between whole managers and agents.

IV. AUTONOMOUS MOBILE ROBOT

Figure 9 shows the test bed of an autonomous underwater mobile robot "The Twin-Burger I." The Twin-Burger I is a fully autonomous robot designed as a versatile test bed for software development. To cover a variety of intelligent behaviors, the robot carries a powerful computer system and multiple sensors to get information on its surroundings. In order to make the robot in small size for convenience of handling, its depth rating and energy duration were determined to be enough for the experiments in a testing tank or shallow water areas.

The Twin-Burger is an open-frame-structured robot which measures approximately 1.5 [m] long and weights 120 [kgf] in air. Most of the on board instruments including the computer system, sensors, and other electronic circuits are mounted in twin FRP pressure hulls. A battery cylinder and FRP hulls are attached to the frame getting high separation between the center of gravity and that of buoyancy, which causes good static stability in rolling and pitching.

The robot is propelled by five thrusters in four degree-of-freedom control modes such as surging, swaying, heaving, and yawing. Position of the robot is calculated by dead reckoning navigation based on the data measured by propeller-type speed sensors. A depth sensor, and a small sensor unit called AHRS (Attitude and Heading Reference System). The robot carries multiple sensors for information acquisition from the surroundings, such as eight channels of ultrasonic range finders and a CCD camera with a tilt-pan mechanism mounted on the front cap of the cylinder. With the range finders, the robot is able to get relative position to the surroundings and to find obstacles which should be avoided. The CCD camera allows the robot to perform vision based behaviors, such as observation of underwater structures, navigation with landmarks, etc. The alignment of these environmental sensors are shown in Fig. 10.

V. EXAMPLE DEMONSTRATION

In this chapter, an experiment using a synthetic world is presented. The experiment was carried out using the Twin-Burger I and the MVS in a small testing pool at the Institute of Industrial Science, the University of Tokyo in June, 1995. The pool has a rectangular parallelepiped shape, and the dimension is (x,y,z) = (10, 5, 1.5) [m] as shown in Fig. 11 (a). Fig. 11 (b) shows a virtual world which consists of four elliptic cylindrical obstacles and a ghost of the Twin-Burger. Fig. 11 (c) shows an image of doing the experiment which the Twin-Burger is swimming in the synthesized testing pool. In the figure, the Twin-Burger which can be seen in the middle navigates from the left to the right avoiding both the actual pool-walls and the virtually-created obstacles. It should be noted that Fig. 11 (c) is the post-processed image for the convenient to explain the settings of the experiment. So the robot did not see this visual image directly but it only sensed the status of the environment through the ranging sonar data represented by the thin arrow-pointed lines in the figure.

The Twin-Burger was connected to the MVS through a high-speed serial communication link during the test as shown in Fig. 12. The transmission rate of the communication is 10 Hz, but the sampling rate of the ranging sonar in both actual and virtual world is 1 Hz. The results of the robot's trajectory and some representatives of the range data are shown in Fig. 13 (a) and (b), respectively. The outer rectangular shape of Fig. 13 (a) (i.e. $x=\pm5$, $y=\pm2.5$) represents the actual wall of the pool. Some parts of the elliptic shapes which are drawn in Fig. 13 (a) represent the virtually created obstacles. In Fig. 13 (b), the heavy gray lines represent the actual range data measured by the Twin-Burger, and the thin solid lines represent the synthetic range data. It can be seen that the synthetic range data are created based on the actual range data, and sometime the virtual range data are chosen when the virtual obstacles are sensed closer than the actual pool-walls.

VI. CONCLUSIONS

In this paper, we proposed a method for the research of underwater robots using technique of "synthetic world" combining virtual world with real world. In this world, the robots behave as if they were swimming in the underwater-world even in the case they are deployed in a simple shaped testing-pool. By testing in the synthetic world, the efficiency of underwater systems' development was improved because the software can be developed directly on the embedded computer system and the hardware and software cross-checking can be easily conducted.

An experiment that an underwater robot navigates in a synthetic world was demonstrated in a small pool with four virtual elliptic cylindrical obstacles and an actual pool. The synthesis was carried out using the ranging sonar with which the actual and virtual Twin-Burger was equipped. The actual Twin-Burger navigated in the synthetic world successfully with avoiding both actual pool-walls and virtual obstacles. Although the setting of the synthetic world created in the example of this paper is simple, much more complex synthetic world can be created easily with the same computational environment.

With the proposed research method, the efficiency of the development of the AUV like robotic systems can be improved. The testing in the synthetic world can be a powerful method for developing more complex, bigger scale underwater robotic systems.

ACKNOWLEDGMENTS

A part of this work is sponsored by JSPS Research Fellowships for Young Scientists.

REFERENCES

M. Bajura, H. Fuchs, and R. Ohbuchi, "Merging Virtual Objects with the Real World: Seeing Ultrasound Imagery within the Patient", J. Computer Graphics, Vol. 26, No. 2, pp. 203-210 (1992)

Y. Kuroda, T. Ura, and K. Aramaki, "Vehicle Control Architecture for Operating Multiple Vehicles", Proc. of 1994 Symposium on Autonomous Underwater Vehicle Technology, pp.323-329 (1994)

Y. Kuroda, K. Aramaki, T. Ura, T. Sugano, and H. Yamato, "Underwater Environment Simulator for Underwater Robots Based on Distributed Processings in Computer Networks", Journal of The Society of Naval Architects of Japan, Vol.178, pp.667-674 (1995), in Japanese

K. Aramaki, Y. Kuroda, T. Ura, and H. Kawano, "MVS: The Virtual Underwater Environment Simulator for Underwater Robotics", "SEISAN-KENKYUU", transactions of Institute of Industrial Science, University of Tokyo, Vol.47, No.12, pp.621-624 (1995), in Japanese

T. Fujii, T. Ura, Y. Kuroda, Y. Nose, and H. Chiba, "Multi-Sensor Based AUV with Distributed Vehicle Management Architecture", Proc. of 1992 Symposium on Autonomous Underwater Vehicle Technology, pp.73-78 (1992)

T. Fujii, T. Ura, and Y. Kuroda, "Mission execution experiments with a newly developed AUV The Twin-Burger", Proc. 8th International Symposium on Unmanned Untethered Submersible Technology, pp.92-105 (1993)

T. Fujii, T. Ura, Y. Kuroda, H. Chiba, Y. Nose, and K. Aramaki, "Development of a Versatile Test-Bed 'Twin-Burger' toward Realization of Intelligent Behaviors of Autonomous Underwater Vehicles", Proc. of OCEANS '93, Vol.I, pp.186-191 (1993)

D. Brutzman, "Integrated simulation for rapid development of autonomous underwater vehicles", Proc. of 1992 Symposium on Autonomous Underwater Vehicle Technology, pp.3-10 (1992)

D. Brutzman, "From virtual world to reality: designing an autonomous underwater robot", AAAI Fall Symp. on Applications of Artificial Intelligence to Real-World Autonomous Mobile Robots, pp.18-22 (1992)

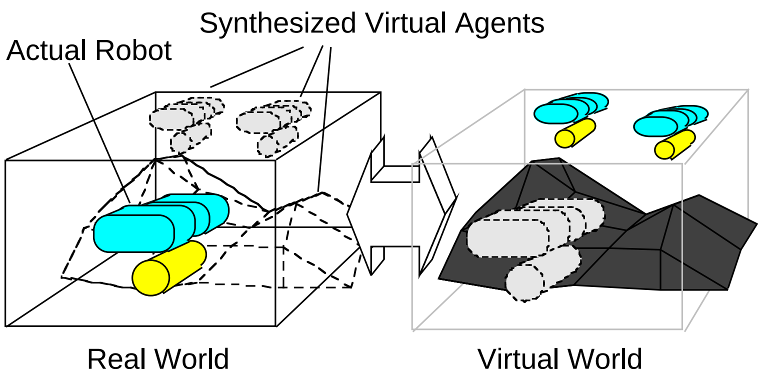


Fig. 1 Creating a synthetic world

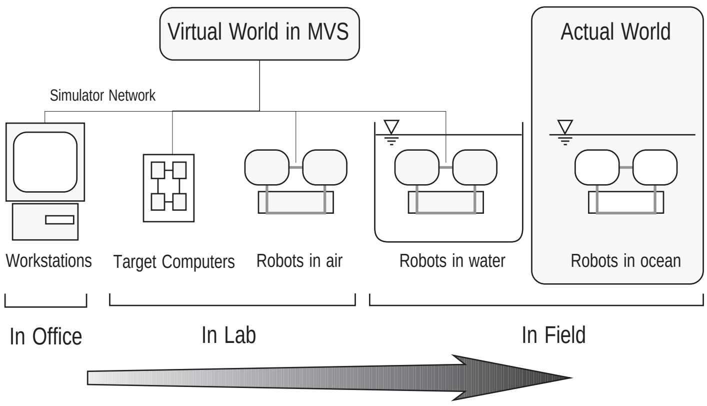


Fig. 2 The integrated research environmet

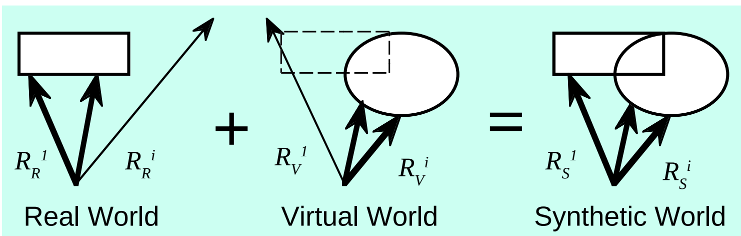


Fig. 3 An example of constructing synthetic sensory data
– in case of using ranging sornar sensors

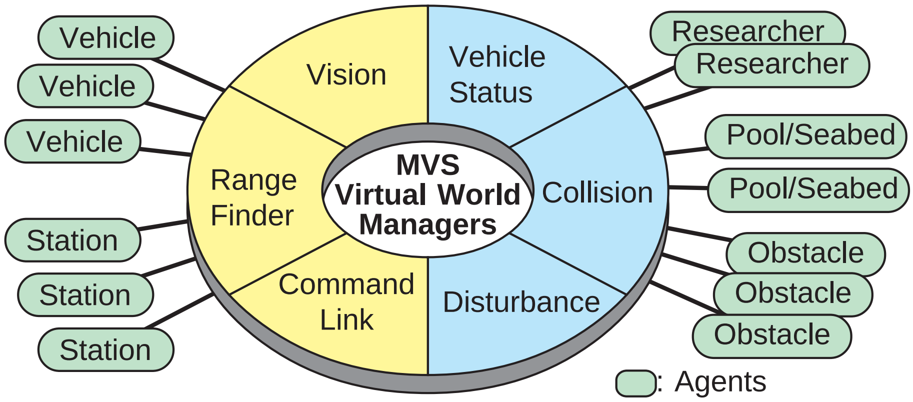


Fig. 4 The distributed architecture of the MVS
- Created virtual world consists of multiple agents

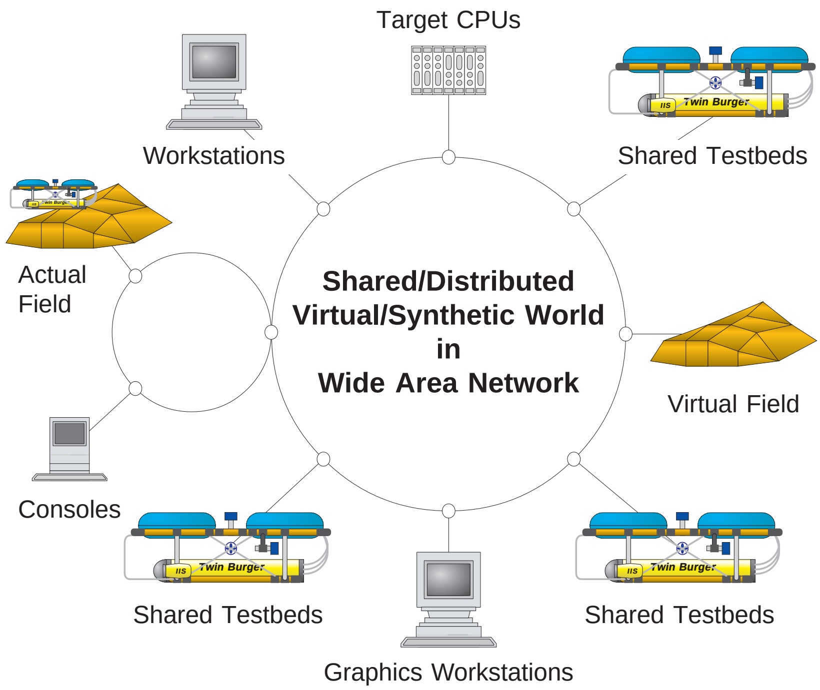


Fig. 5 The large scale simulation in wide area network

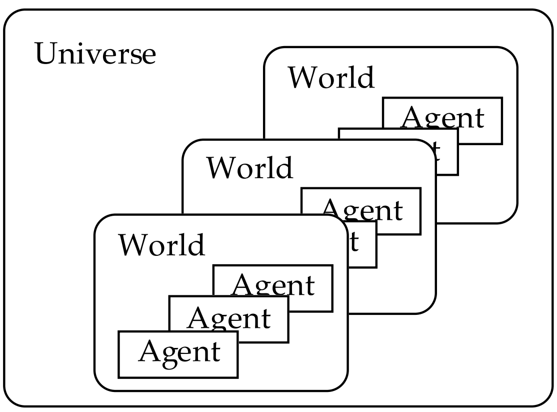


Fig. 6 The "Universe" of the MVS

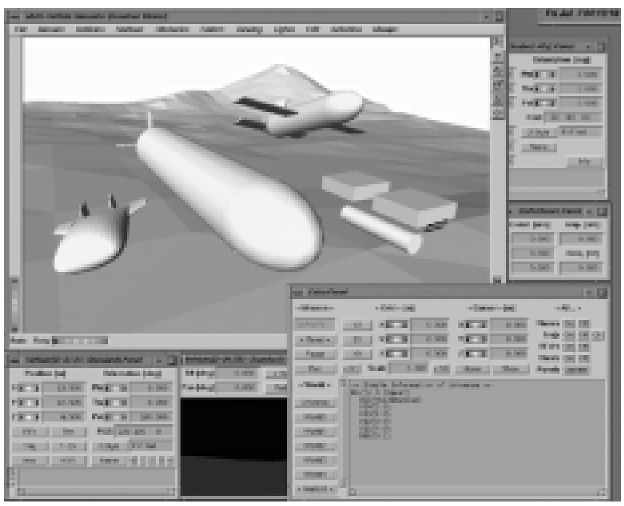


Fig. 7 Graphical user-interface display of the MVS

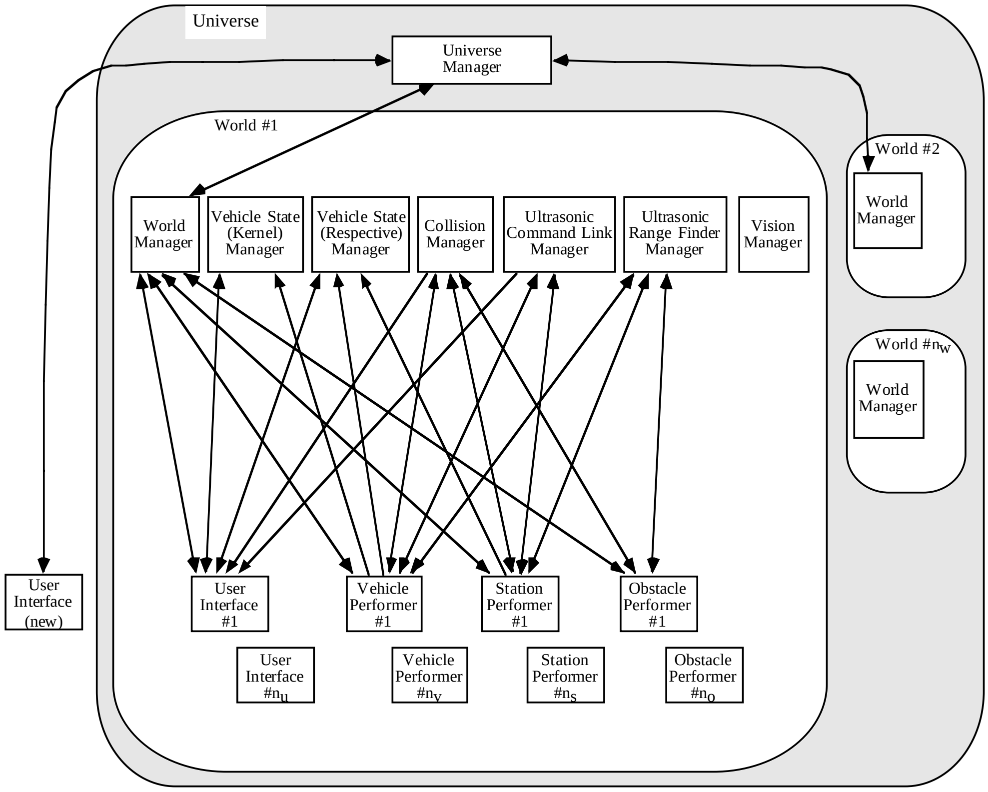


Fig. 8 The software coordination of the MVS

Fig. 9 Twin-Burger 1: the versatile underwater robot testbed

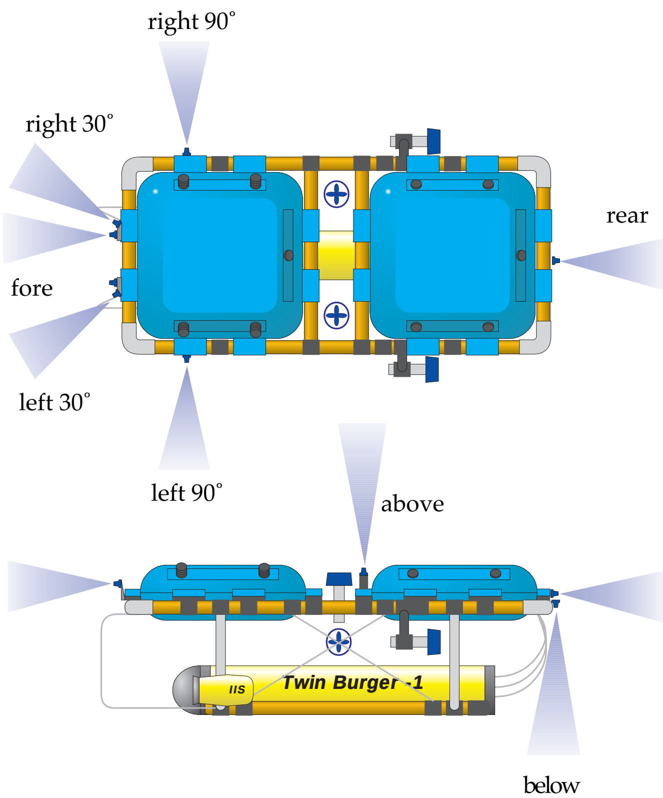
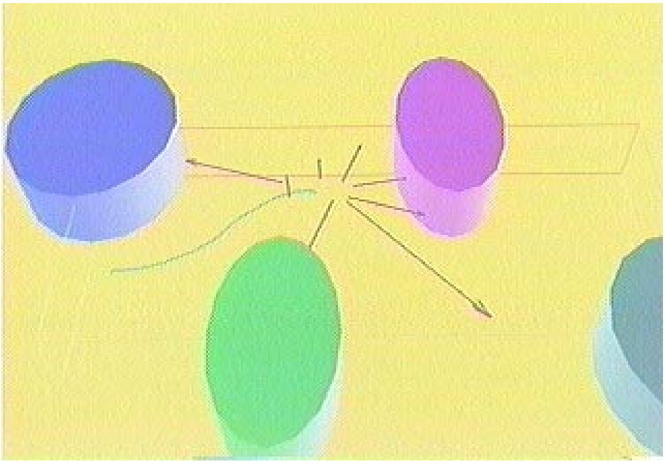


Fig. 10 The alignment of ranging sonar sensor of the Twin-Burger

(a) Actual underwater-world
a simple pool and an actual robot



(b) Virtual underwater-world
four obstacles and a *ghost* of the robot

(c) Synthesized underwater-world

Fig. 11 An actual robot is swimming in a merged world. 1) Arrows pointing to the pool walls (i.e. 90˚ left and 30˚ right of the robot) are visualized images of the actual range data measured by the robot. 2) Arrows pointing to the virtual obstacles (i.e. 30˚ left, fore, and 90˚ right of the robot) are those of the virtual range data transmitted from MVS into the robot in real-time.

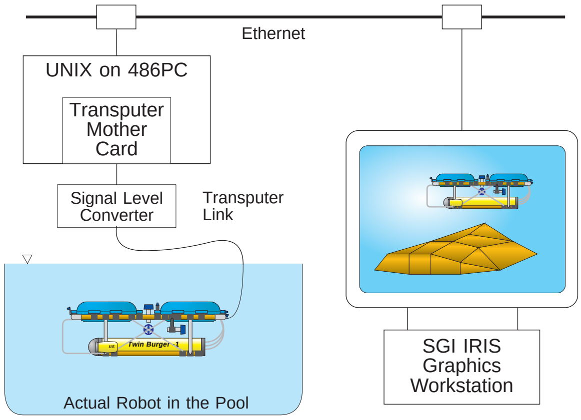
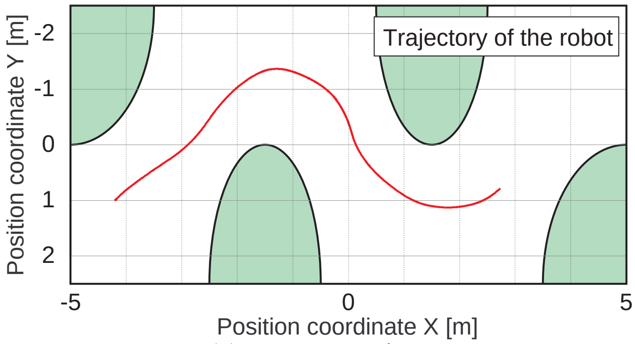
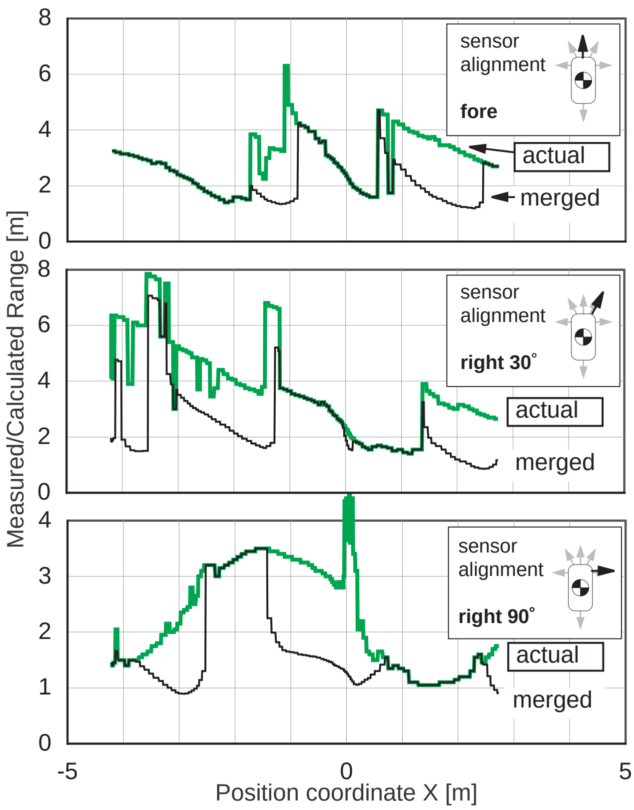


Fig. 12 Hardware configuration of the experiment



(a) trajectory of the robot



(b) some examples of the range data

Fig. 13 Results of an experiment – Actual/Synthetic ranging sonar data (i.e. fore, 30˚ and 90˚ right directions in body coordinate), and the trajectory of the robot in x-y plane