

The underwater vehicle MAUVE: a user-friendly tool for coastal survey

M.-J. RENDAS and Y. HEBRARD

Abstract—This paper describes the autonomous underwater vehicle Mauve, which is a small vehicle designed for the survey of coastal areas. It presents the physical characteristics of the platform, and the modules on which the design is based. It describes in detail the man-machine interface used for programming each mission and recovering and visualising the acquired data, giving in this way a complete presentation of the missions that can be performed by the vehicle.

Index Terms—Autonomous underwater vehicle, underwater robotics, man-machine interface, mission definition.

I. INTRODUCTION

We present the miniaturized autonomous underwater vehicle MAUVE, which is the result of a joint effort of several European partners in the context of two EU-funded projects. This vehicle is targeted for the surveying and mapping of coastal areas: it is light and small, allowing launching and recovery by a single person from a small boat, minimizing in this way the logistics associated to its operation. A special attention has been given to the design of the user interface of the vehicle, which has been conceived with the community of final end-users (biologists, oceanographers,...) in mind: simple graphic input of the mission definition, variety of possible formats for site-specific information, availability of simulation tools for pre-mission validation, and automatic choice of vehicle-related parameters from the high-level user definition of the mission. For development/upgrading purposes, the interface contains a completely separated “expert level” that provides the necessary interface for engineering developments, and which is not detailed here. In this paper, we present an overview of the vehicle, focusing on the presentation of its man-machine interface.

II. PHYSICAL CHARACTERISTICS

A. Shape, size and weight

Mauve is a torpedo shaped AUV with 1,9 meters long and 15 cm diameter, see Figure 1. Its body is made of carbon fiber, with some sections in aluminium, and a rubber nose that houses acoustic equipment.

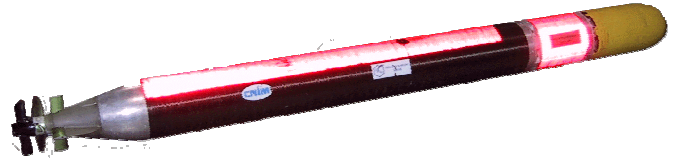


Figure 1: MAUVE.

It weights about 35 Kg in the air, and has a positive buoyancy of about 1 Kg in the water, allowing it to be easily operated by a single person from a small boat, and automatic surfacing when the motor is switched off.

B. Power, propulsion and steering

Mauve is powered by one lithium battery, providing an endurance of about 3 hours at its nominal speed of 2 meters/sec. This is sufficient for the small coastal missions for which it is presently used.

All propulsion is at the rear, being provided by a DC motor. The vehicle can be steered in the horizontal plane by a rudder at the rear, and in the vertical (diving/pitch control) by a single fin placed also at the rear.

III. SUB-SYSTEMS

The complete Mauve system comprises the vehicle itself,

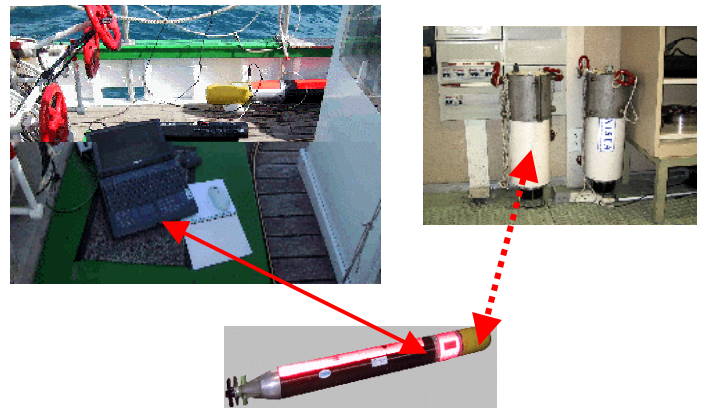


Figure 2: physical sub-systems.

two external acoustic transponders for localisation and a portable PC for human interface.

The portable PC communicates with the vehicle through a

serial link before and after mission, for downloading mission configuration and recovery and display of the acquired data. This man-machine interface is described in detail in the rest of the paper.

The two acoustic transponders, that communicate with the vehicle during mission through a spherical transducer installed in the vehicle head, must be located at calibrated positions prior to vehicle launching. They have several operational modes, the most basic one being to act as responders to the vehicle queries, and allowing thus vehicle positioning during mission execution. Their nominal range is 3 Km, defining in this way the autonomy of the vehicle (excluding dead-reckoning excursions behind this range). The distance data derived from this equipment is fused with on-board navigational data by an Extended Kalman filter that keeps updated estimates of the vehicle position, attitude and velocities and sea current.

IV. ON-BOARD MODULAR ORGANISATION

The vehicle itself comprises several modules, see Figure 3. Propulsion and power are located at the rear of the vehicle, together with the Unit Control Vector, a micro-processor controlled board which performs basic sensor acquisition and implements the low-level control loops, acting directly on the motor input and controlling the two navigation fins. This micro-processor also implements basic security checks, and may decide to abort the mission (triggering surfacing) if the valid operational range of a number of key parameters is violated.

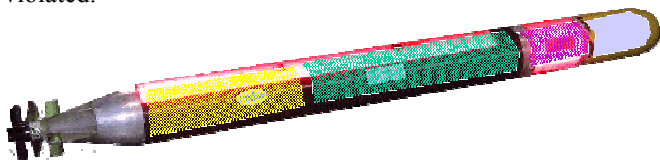


Figure 3: on-board sub-systems

The Acoustic Processing Board located at the front of the vehicle comprises the spherical acoustic transducer, and electronic boards for signal conditioning, pre-filtering, signal processing, and communications control.

The user payload is located in a central section, close to the acoustic equipment. It contains the user sensors. It is in this section that the external plug for programming and data uploading is located.

All other subsystems communicate directly with the Mission Management and Navigation System of the Vehicle (MMNS), which is the central computer on-board. This module is responsible for sequencing the mission. It integrates all positioning, guidance and high-level control software.

V. MAN-MACHINE INTERFACE

The man-machine interface of Mauve has three main functions: enable simple and graphical definition of the mission to be performed by the vehicle; transfer the mission to the vehicle before launching; and upload and visualize acquired data.

The interface is organized around the notion of “project”, which gather all data related to a given mission: robot configuration, site related data, auxiliary sub-systems configuration, the tasks to be executed by the vehicle, and folders of uploaded data.

The graphical part of the interface is coded in Tcl/Tk, while all computational operations are coded in C++ (dynamically linked DLLs). It runs on a standard portable PC under Windows 98 or higher.

The subsequent sections of the paper present this interface, giving examples of the screens displayed, and the information asked from and provided to the user.

VI. THE PROJECT MENU

When the application is started, by mouse clicking on the *Ihm* icon, only the **Project** button of the interface is enabled, see Figure 4. The user must use it to either open an existing project or create a new one (in this case he is prompted to enter identification information: name and date). Figure 4 shows the dialog box for opening an existing project. If the user chooses to open an existing project he can browse the interface’s “projects” directory and select one by mouse-

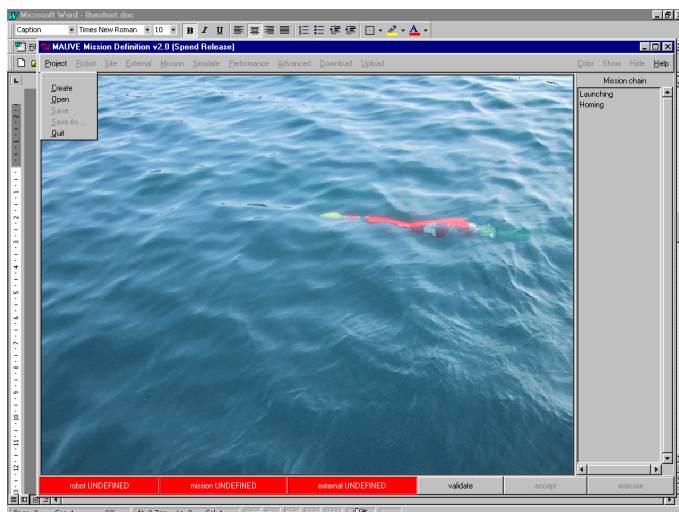


Figure 4: Project menu.

clicking. All modifications to the opened project can later be saved, or a new project created by saving the current one under a different name. When the project is completely specified (which is signalled by the fact that all bottom bar buttons become green, see for instance Figure 7) all the buttons of the top tool bar become enabled.

VII. THE ROBOT MENU

Once a new project name has been correctly specified, the buttons **Robot** and **Site** of the tool-bar of the main window of the interface become enabled, see Figure 5, and the project name (in the example, “test_create”) is updated at the title bar of the window. Clicking in the button **Robot** the user is offered three following choices: “Edit” or “Save as”. By clicking on “Edit,” he can enter information concerning the

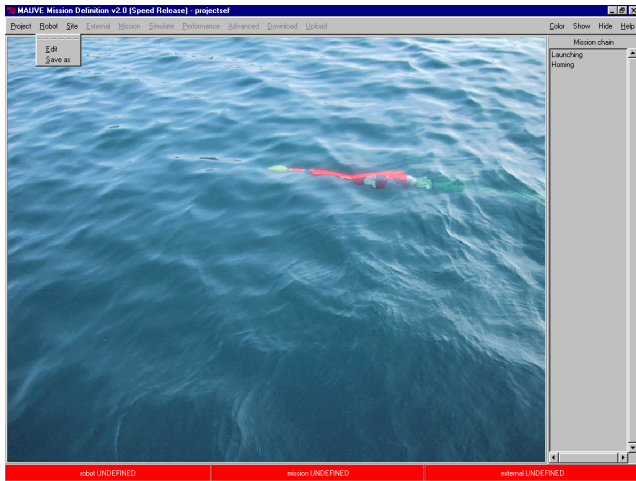


Figure 5: Robot menu.

vehicle, see Figure 6. A set of default values (designed for the present configuration of the Mauve vehicle) are already pre-defined, for use by an un-informed user.

In the Robot dialog box, two kinds of sensors are distinguished: payload sensors on the left and navigation sensors on the right. This box allows also to specify the minimum, maximum and nominal speed for the mission (in

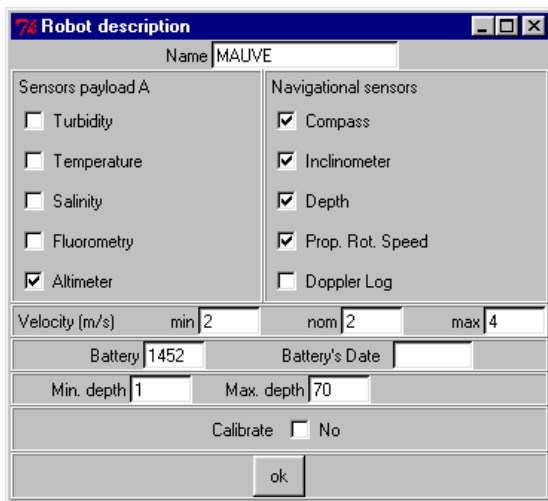


Figure 6: Robot dialog box.

meters/second) and the minimum and maximum depths that will be imposed for the mission. Note that the range of these variables are limited reflecting the hardware limitations of the platform (the valid intervals can only be modified under the “expert” level). This prevents an un-informed user of launching a mission that cannot be safely performed by the robot. The flag calibrate indicates whether or not the robot will execute pre-defined manoeuvres to calibrate its compass bias at mission start (the default value should be kept by un-informed users).

By clicking on the “Ok” button in Figure 6 the robot configuration becomes defined, and the corresponding bar at

the bottom of the interface screen becomes green, and its text is changed to “robot defined”, as in Figure 7.

By clicking on the “Save as” choice in the robot menu, the current robot configuration is (by default) saved in the file “robot.cfg” in the project directory. The user can choose to save the information in a different file, but this is the file that will be read when the project is open.

Note that all the information concerning a project is either globally saved using the choices “Save” or “Save as” in the Project menu, or individually saved in each menu using the corresponding “Save as” choice in each menu.

VIII. THE SITE MENU

Clicking on the button **Site** pops-up a window that enables the user to enter information concerning the operating site, see Figure 7.

Clicking on “Read” allows copying the site configuration of another project, by indicating the location of a “site.cfg” file. The user is prompted with a file selection window, that he/she can use to browse the projects directories to select the project’s site configuration file that he/she wishes to use.

Clicking on “Edit” will pop-up the dialog box shown in Figure 8.

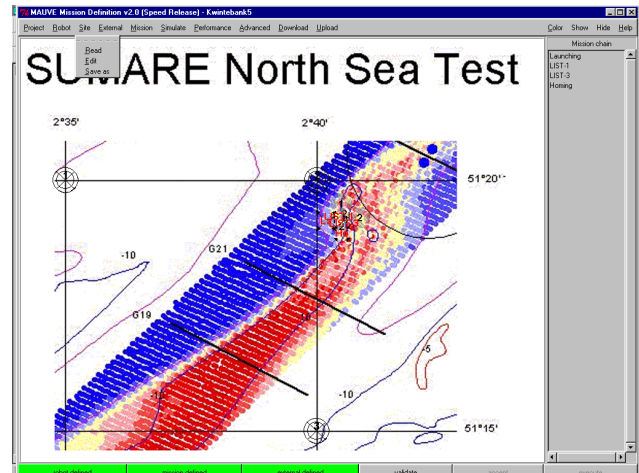


Figure 7: Site menu.

The user can enter site information in three distinct ways:

1. By giving the name of a “.map” file (this is an internal format of the interface, gathering the name of the figure that is displayed on the main canvas, and the geographic coordinates of three corners).
2. By manually defining a coordinate system on the main canvas.
3. By providing the name of a file with a grid of bathymetry values, as shown in Figure 9. An example is shown in Figure 14.

In the second case, the user is given two options:

1. to specify an image of the operating site (e.g. scanned nautical map), enabling easy definition of all subsequent mission parameters. The user must in this case manually enter the coordinates of three non-collinear points in the image, to define a coordinate system upon the screen.

Figure 7 shows an example of a screen obtained with a digital image. The points used to define the coordinate system can be seen at the intersection of the latitude and longitude lines.

- alternatively, a mission can still be defined on an empty canvas, on which a coordinate system is imposed by specifying the coordinates of three non-collinear points.

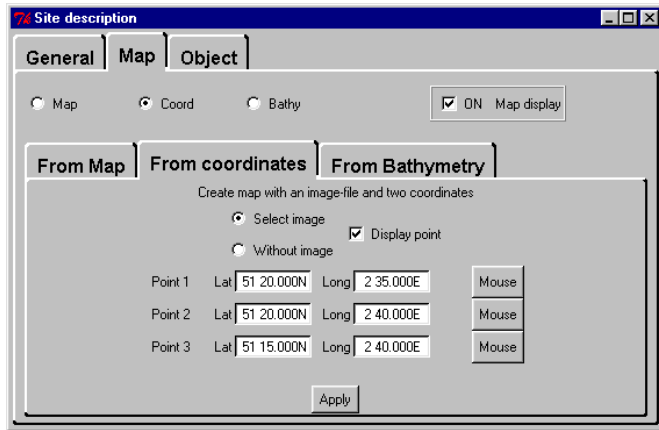


Figure 8: Site definition window.

In the case a bathymetry file is specified, see Figure 9, the user is requested to enter the name of the file, the size of the grid (in the example 100 meters) and the filename for the corresponding gif file that will be automatically created. Site information is internally updated by pressing “Ok” in the “General” page of the Site dialog box. The information can be saved to the project’s directory site configuration file “site.cfg” (or to a distinct file, see comment on the section on Robot configuration) using the choice “Save as” in the general Site menu of the interface.

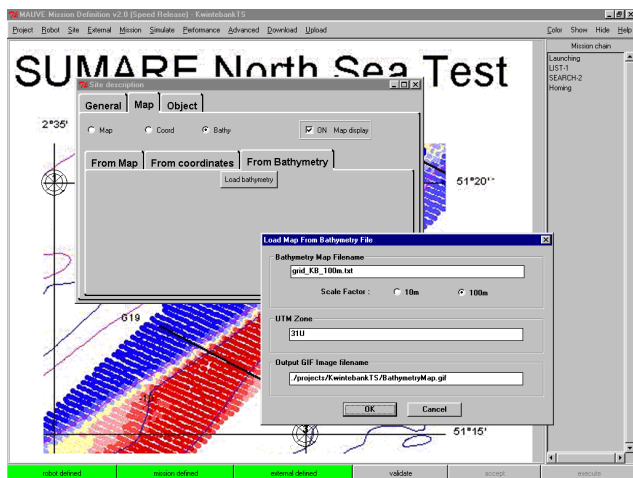


Figure 9: specifying a bathymetry file.

The interface offers also facilities for zooming, color selection and choice of the displayed elements.

IX. THE EXTERNAL MENU

Once both the robot and the site are conveniently configured, the user is given access to the **External** menu, that enables definition of

- Launching point** (latitude, longitude, error). The error parameter describes the confidence on knowledge of the position of the launching point (precision of GPS receiver,...). The user can also enter information concerning the direction used when launching the vehicle.
- Homing point** (latitude, longitude, depth, error). The depth parameter is the depth that the vehicle will use when returning to the homing point upon mission completion. The error parameter sets an upper bound of the vehicle’s positioning error when returning to the homing point. This security will be enforced throughout the mission, and it may cause anticipated mission suspension if it cannot be guaranteed. This security is provided to minimize problems at recovery, and the value entered must be adjusted to the visibility conditions at sea: smaller errors can be tolerated in rough sea.
- Position of the two *acoustic beacons* (latitude, longitude, depth and error). The error parameter describes the confidence on the knowledge of the beacon positions, usually corresponding to the knowledge of the precision of the GPS receiver used to calibrate their position. It is used on-line to tune internal filters of the real-time positioning sub-system of the vehicle, as well as for the simulation of the mission. A flag button indicates whether initial *auto-calibration* steps must be performed before entering nominal mission execution. This flag is, by default, disabled, and the non-informed user should not change its value.

All latitude and longitude values can be entered either by mouse clicking on the screen canvas, or by directly typing their latitude and longitude.

X. THE MISSION MENU

The **mission** menu allows the specification of the vehicle trajectory and of the sensors that must be recorded during each step of the mission. A mission is a sequence of basic subtasks. At the moment the only tasks that are fully implemented are: Go to, List of points, Search, 2D Grid, and Open Loop.

The mission being defined appears as a list in a frame at the right of the screen, see Figure 15. The user can at any moment change the order of the subtasks (by mouse dragging), delete, or modify any of them (by double clicking on them).

Figure 10 shows the dialog box allowing the parameterisation of Go-to tasks. The user must enter the target point, either by manually entering its geographical coordinates or by mouse clicking on the main canvas and by manually entering its depth (in meters). The user specifies the speed of the vehicle when traveling to the target point (in meters/second). By pressing the “Delete” button, the task is deleted from the mission list (the end point of the previous task automatically becomes the starting point of the next task).

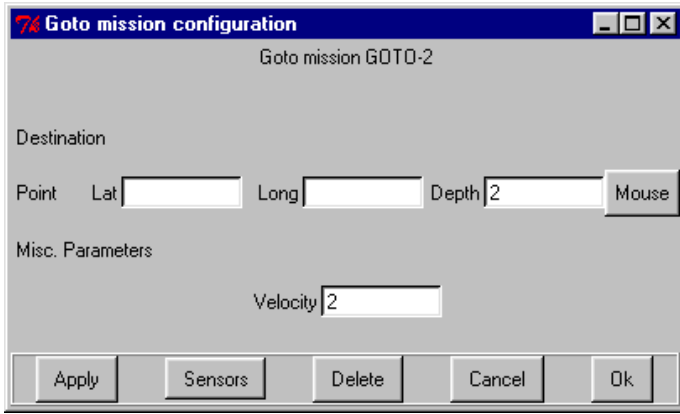


Figure 10: Goto missions.

Figure 11 presents the dialog box for List tasks, which are a

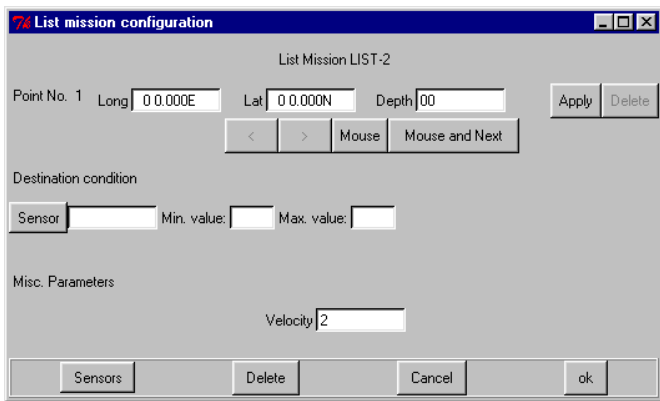


Figure 11: List tasks.

chain of basic “Go-to” ’s. Using the buttons “<” and “>” the user can access the previous and next points in the list. He/she can use these buttons, together with the “Mouse” button, to specify each point in the list, or else use the “Mouse and Next” button that automatically adds a new point at the end of the current list of points. Note that the depth of each point must be

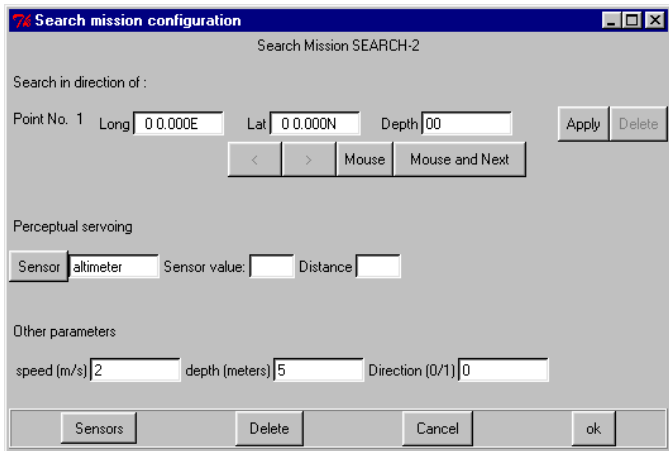


Figure 12: Search tasks.

manually entered (in meters) as well as the speed used in each

segment. Otherwise the minimum allowable vehicle’s depth and the nominal speed are automatically used. This dialog box allows the user to specify an additional test to be continually performed on a sensor value. The user must specify which sensor he/she wishes to test, and the condition that must be verified during the segment (interval of accepted values). If this condition is violated, the robot will switch to the next point in the list. This feature has been defined to allow mapping of sensor-defined regions (plumes,...) and is not yet fully implemented, and is not currently exploited by the real-time management system of Mauve.

Figure 12 shows the dialog box for Search tasks. Initially, this dialog box has been designed to allow search of a given condition/object inside a polygonal area: the user specifies a list of points (in the same manner as for the “List” tasks) that defines the contour of a closed polygon at the interior of which search will be performed. It has been recently adapted to test perception driven (contour tracking) guidance. The current implementation allows only the use of one point, leading to a search along one line, until the specified condition (in the example, a given value of the altimeter readings) is observed, which is then tracked during a given distance. The nominal starting point and end point of these tasks are the same: if the sensor condition is never met before the target point is reached, the vehicle returns to the initial position; otherwise, at the end of the specified tracking length, the vehicle reverses its path; the sensor condition is tracked in the opposite direction for the same duration, and the vehicle returns to the starting point.

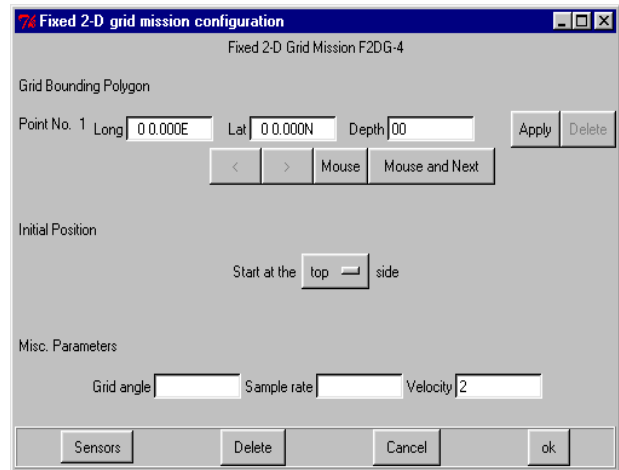


Figure 13: Grid tasks.

Figure 13 shows the dialog box for “2D Grid” tasks. The user defines a closed polygonal region (as for the “List” tasks) at a given depth (in meters), and specifies a desired horizontal sampling rate (“Sample rate”). The vehicle will survey the interior of this region using parallel tracks at the direction specified by “Grid angle” using the specified speed (in meters/second). He can also choose the starting point for the survey: “top,” “bottom,” “left” or “right”. The interface will automatically determine the number of parallel tracks to completely cover the specified region.

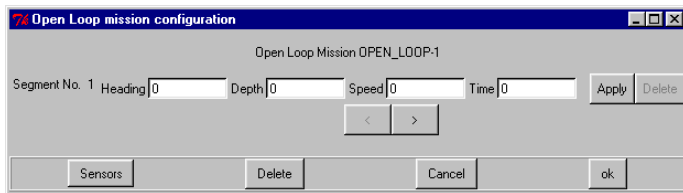


Figure 14: Open loop tasks.

Figure 14 shows the dialog box for “Open Loop” tasks. The vehicle will chain a sequence of “segments” directed at the indicated “Heading”, at the given “Depth” and “Speed” (meters/second), during a given “Time.” This type of basic task has been designed to facilitate testing of the low-level controllers of the robot, inducing a behaviour independent of the high-level guidance modules, and thus allowing direct observation of the stability of the AUV auto-pilots.

XI. THE SIMULATE MENU

When the mission is defined, the buttons **simulate**, **download** and **advanced** are enabled, see Figure 15.

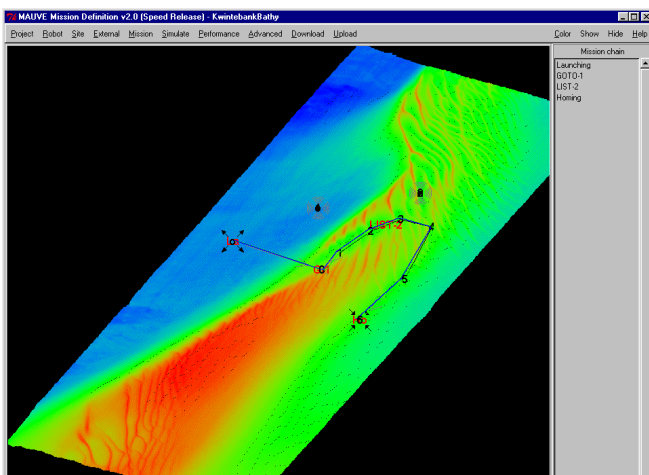


Figure 15: mission simulation

The menu “**Simulate**” enables simulation of the mission, to evaluate the errors that can be expected in the nominal conditions specified. It can be important to help a non-specialist user to tune the position of the acoustic transponders, which determine in a critical manner the non-homogenous positioning error pattern. This module relies heavily on the dynamic model that has been identified for the vehicle.

Figure 15 shows the interface screen at the end of a simulated mission, consisting of one go-to task and one list task. The “real” and “estimated” trajectories are shown in green and blue. The position of the acoustic references can also be seen in the region North of the vehicle’s trajectory.

XII. THE DOWNLOAD MENU

The menu **download** is used to transfer the mission definition to the vehicle, through a serial RS 232 link., offering two choices: “Mission Parameters” and “Data Visualization”. If

the “Mission Parameters” button is clicked, the user is prompted with a window summarizing the mission and configuration parameters defined. If the user accepts the current mission definition, he/she can start actual download to the vehicle by pressing “Download.” The user must then make the following operations (by that order)

1. make sure that the serial connection is established,
2. press the “Download to MMNS” button on the dialog window (see Figure 16)
3. turn on the vehicle power by using a magnet

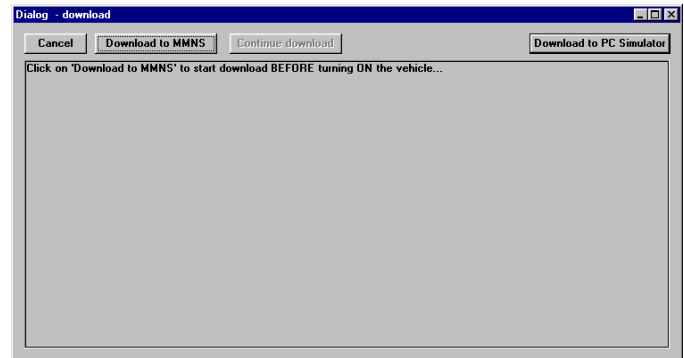


Figure 16: Download Dialog box.

Information concerning progress of the mission download is given to the user. If persistent problems are detected, the user is informed, asked to verify that the physical connection is established and the vehicle powered and re-start download, or else abandon download by pressing “Cancel”.

If successful transfer of all mission definition and configuration parameters is achieved, the user can trigger vehicle launch (or otherwise abandon the entire procedure). If the launch order is accepted, the user is informed that the vehicle is ready to be launched, by unplugging the serial link at the vehicle’s nose.

The button “Download to Simulator” in Figure 16 is used for “hardware-in-the-loop” table simulation of a mission, and triggers communication with a separate PC that simulates the dynamic behavior of the vehicle and communicates with the actual MMNS and UCV boards by simulating all the sensor readings and actuators of the vehicle. This facility is extremely useful for technical development and testing before sea-trials, but is not required for the normal user, and it is not described here.

XIII. THE UPLOAD MENU

This button offers two choices: acquire mission data and visualize previously acquired data.

A. Data acquisition

The first choice must be used to recover the log files of the distinct sub-systems. The current version of the interface sequentially uploads all these log files. Current work is being done on implementing two distinct uploading phases: one for the general user (where only the log files required to visualize the user requested data will be uploaded) and an expert level,

which will allow recovery of internal historic files for engineering development.

Two windows are shown to the user: an output window, that reports on progression of uploading, and a dialog box that allows the user to save the acquired files in a separate directory under the project's directory. This allows saving of distinct realizations of the same mission without requiring manual copying of the recovered files.

As for download, the user must execute the following steps **by order**:

1. make sure that a connection is established
2. press the "Start Upload" button
3. turn power on using a magnet.

As for the download menu, the user is informed of the progression of the upload procedure. If complete upload of all log files is achieved, the user is invited to use "show graphics" on the **Upload** menu to visualize the acquired data. If he clicks on this button, the user is presented the dialog window shown in the Figure 17 below.

B. Data visualization

By mouse clicking on the different buttons shown in Figure 17, the user can select the log file that he/she wishes to examine. There are presently four log files that are retrieved from the

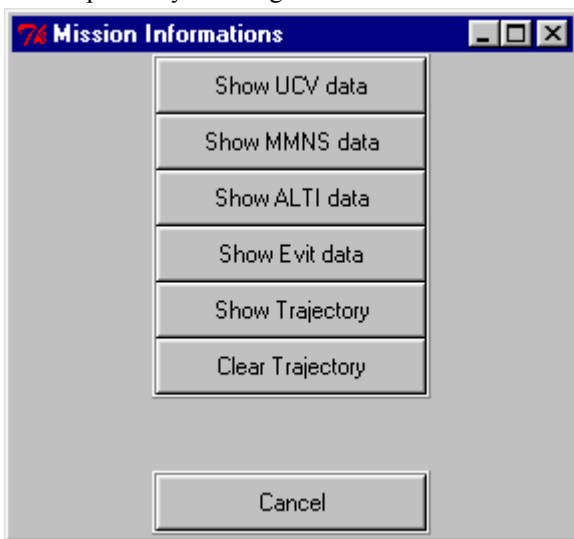


Figure 17: Visualisation options.

vehicle:

- the log of the Mission Management and Navigation System, which gathers, versus time, sensor readings, position and velocity estimates, and references sent to the UCV auto-pilots. Using the mouse, the user define the set of entries that will be plotted, by setting the corresponding flags, see Figure 18.
- the log file of the altimeter : it records, versus data, the 10 altimeter readings received every second.
- the log file of the avoidance sensor, with the same format as the altimeter.
- the log file of the UCV. It records all sensor values, commands generated and error flags. This file is important for debug only.

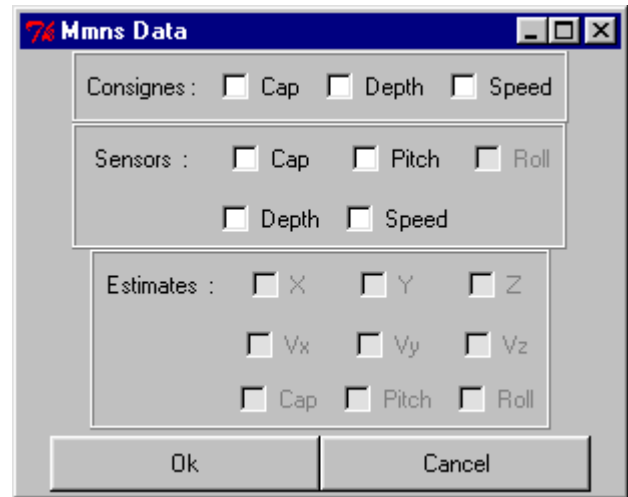


Figure 18: Ploting options.

- The two additional buttons "Show trajectory" and "Clear trajectory" in Figure 17 allow drawing and erasing of the vehicle trajectory on the canvas of the interface, leading to plots similar rto those obtained for the simulator, see figure 15.

ACKNOWLEDGMENT

This work has been partially funded by the European Union through projects MAUVE (MAST program) and SUMARE (IST program.). Information about the project SUMARE can be found at <http://www.mumm.ac.be/SUMARE>. The authors acknowledge the collaboration of the other partners of these projects, in particular engineers from Thales Underwater Systems (former Thomson Marconi Sonar) and members of MUMM, in defining this interface. The current software is the result of a team work at I3S, and integrates the contribution of several students from the engineering schools ESSI and ESINSA (University of Nice Sophia Antipolis), who did their graduation project in the mobile robotics team of I3S.