PAPER

Unmanned Surface Vehicle for Coastal and Protected Waters Applications: the Charlie Project

AUTHORS

M.Caccia
M. Bibuli
R. Bono
Ga. Bruzzone
Gi. Bruzzone
E. Spirandelli
Consiglio Nazionale delle Ricerche
Istituto di Studi sui Sistemi Intelligenti per l'Automazione

ABSTRACT

The paper deals with the design, development, test and exploitation of an unmanned surface vehicle (USV) for coastal and protected waters applications, such as sampling of the sea surface microlayer and harbor survey. The unmanned catamaran Charlie, which is fully autonomous from the point of view of both power supply and computing capabilities and is supervised by a portable human operator station, is characterized by a state-of-the-art hardware, software and control architecture. A PC-compatible computing platform and a GNU/Linux-based embedded real-time system support an intelligent control architecture, while advanced modeling and identification techniques, based on on-board sensors, allow the implementation of accurate navigation, guidance and control model-based algorithms. Results of experiments carried out on motion estimation and maneuvering control in the Genoa-Prà harbor are presented and discussed as well as the vessel exploitation in scientific applications and research projects.

1. Introduction

In the last years, a number of unmanned surface vehicles (USVs) have been developed for bathymetric surveys, environmental monitoring and sampling, coastal defense, and supporting of Autonomous Underwater Vehicle (AUV) operations. An example is given in the development of a family of USVs at the MIT AUV Lab (Manley et al., 2000) since the early 1990's. ARTEMIS, a small-scale replica of a fishing trawler, used for generating bathymetric maps by executing DGPS waypoint-defined surveys supervised by a human operator through a radio modem, was followed by the catamarans for automated bathymetry ACES (Autonomous Coastal Exploration System) and its evolution AutoCat, based on the commercially available Float Cat™ hulls. In the last years, to deal with the legal issues related to the operation of USVs at sea, basic trials of in-field autonomous operation of unmanned marine vehicles in accordance with conventions for safe and proper collision avoidance as prescribed by the Coast Guard Collision Regulations have been executed by using a couple of SCOUT (Surface Craft for Oceanographic and Undersea Testing) autonomous kayak test platforms developed by MIT (Benjamin et al., 2004). On the other side of the Ocean, the Lisbon IST-ISR Dynamical Systems and Ocean Robotics Laboratory is carrying out the development of a flotilla of autonomous marine vehicles such as Delfim, a small autonomous catamaran for testing the concept of an autonomous surface craft capable of working in close cooperation with an autonomous underwater vehicle (Pascoal et al., 2000), and Caravela, an autonomous robotic research vessel capable of performing a large number of oceanographic missions without support from a permanent, dedicated crew. In Europe again, the University of Rostock developed Measuring Dolphin for carrying measuring devices in the field of oceanography, water ecology and hydrology (Majohr et al., 2006), while the University of Plymouth developed Springer, an autonomous catamaran designed to sense, monitor and track water pollution (Xu et al., 2006). As far as military applications are concerned, the main operational advance has been given by the QinetiQ Ltd shallow water influence mine sweeping system (SWIMS) (Cornfield et al., 2006), employed by the Royal Navy to support mine countermeasures operations in Iraq in 2003. SWIMS basically consisted of the development of a conversion kit to transform existing Combat Support Boats, already operated by the British Army, in remote controlled vessels. Other military USVs such as the testbed developed at SSC San Diego (Ebken et al., 2005), based on the Bombardier SeaDoo Challenger 2000, and the Israeli Protector USV focus research and development efforts in the integration of sensors for over-the-water obstacle detection and avoidance as, for instance, radar, electro-optic sensors and machine vision technologies.

In this context, the Charlie project focuses on the evaluation of the fundamental issues in autonomous navigation, guidance and control of surface craft in coastal or protected waters on the basis of the requirements given by applications such as sea surface microlayer sampling or harbor surveying.

The project also addresses more general issues, strictly related to the overall research activity in the field of marine robotics and industrial automation carried out by the *Autonomous robotic systems and control* group of CNR-ISSIA. This is, for instance, the case of the design and implementation of a platform for robotics and manufacturing based on commercial-off-the-shelf (COTS) hardware and free software, of the development of intelligent control architectures for mobile robots, and of the design and implementation of advanced navigation, guidance and control systems.

The Charlie project started in 2002 on the basis of a specific operational need: to develop a semi-autonomous vessel for facilitating the collection of sea surface and immediate sub-surface water samples in Antarctica. The result was a prototype catamaran designed to maneuver at very low speed in calm water, that, after its satisfactory exploitation during the Italian Antarctic expedition 2003-2004 (Caccia et al., 2005a), was enhanced with the integration of a rudder-based steering system and a new software control system fully based on standard GNU/Linux. The resulting USV can work as a testbed for modeling, identification, navigation, guidance and control research towards the development of a vessel able to fully autonomously operate in coastal and protected waters in the presence of usual traffic.

The paper discusses the steps in developing a research prototype robotic vessel: the electro-mechanical architecture (section 2); the embedded real-time software infrastructure and control architecture (section 3); the navigation, guidance and control algorithms and modeling and identification techniques (section 4). The motivations of the architectural and algorithmic choices as well as the lessons learned during the progress of the project are discussed with the support of experimental results. A brief description of the projects exploiting the Charlie USV is given in section 5. The paper concludes with the discussion of the basic technological and research issues to be faced for further system development.

2. Electro-Mechanical Architecture

The system consists of a robotic vessel, with on-board computing resources for navigation, guidance and control, and a portable human operator station for remote control and supervision. Both are autonomous from the point of view of power supply.

As far as the robotic vessel is concerned the catamaran-like shape was chosen, on the basis of previous experience reported in the literature, for stability with respect to roll, redundancy in hull buoyancy, and capability of payload transport with respect to the hydrodynamic drag. In particular, the original requirement of mounting a Harvey-like cylinder for sea surface microlayer sampling (Harvey, 1966) required the design of a catamaran with a space between the two hulls of 0.90 m. The vehicle length of 2.40 m was determined by logistical constraints such as the space available on the stern deck of the Antarctic support vessel and on the CNR-ISSIA truck for transporting instrumentation. The result was a vehicle with a width of 1.80 m, a hull height of about 0.60 m, a weight of about 300 Kg in air, and the possibility of carrying a payload of about 120 Kg. In order to avoid contamination for organic and inorganic samples, the hull was constructed in fiberglass covered with epoxy system resin. A view of the Charlie USV on the deck of the support vessel Malippo during the Antarctic mission in February 2004 is shown in Figure 1.

Since for the sampling of sea surface microlayer the vehicle had to operate essentially at low speed, and commercially available electrical thrusters for small boats do not usually guarantee a fine velocity tuning at low propeller revolution rate, Charlie USV was equipped with Romeo ROV's actuators (Caccia et al., 2000a), specially designed for operating near bollard conditions. Each thruster consists of a DC motor (330 W at 48 V) with corresponding tachometer and rotary joint and is coupled with a three blade propeller. A set of servo-amplifiers perform proportional-integral differential (PID) control of the thruster velocity on the basis of the error between the signal of the motor tachometer and the reference speed.

In the new release of the vehicle, the steering system, originally based on the differential propeller revolution rate for working basically at very low speed, has been integrated by two rigidly connected rudders mounted behind the propellers. The rudders are actuated by a brushless motor driven by a standard motion controller able to handle out-of-range signals. Magnetic and mechanical switches generate suitable signals in order to keep the rudder angle module lower than 45 degrees and to automatically locate the zero position when the system is turned on. Figure 2 shows the Charlie USV equipped with the rudder system.

Power is supplied by a serial package of four lead batteries (42 Ah at 12 V each one), integrated with a set of four flexible solar pan-

FIGURE 1

Charlie USV on the Malippo deck. Terra Nova Bay, Ross Sea, Antarctica. February 2004.

FIGURE 2

Charlie USV during NGC trials, Genoa Prà harbor, June 2006. The brushless motor controlling the rudder motion is clearly visible in the centre of the stern.

els (32 W at 12V) for supplying power peaks and increasing system autonomy.

The navigation package consists of a GPS Ashtech GG24C providing geographic coordinates at 1 Hz, integrated with a KVH Azimuth Gyrotrac, able to compute the True North at 2 Hz on the basis the measured Magnetic North and the GPS-supplied geographic coordinates.

As far as the data acquisition and control system is concerned, the mature and consolidated standard represented by industrial PC-derived computers, supporting GNU/Linux and making available a real-time clock (which is a basic requirement for using standard GNU/Linux as an embedded real-time operating system, see section 3), has been fully exploited. Thus, the Charlie computing system consisted of a Single Board Computer (SBC) board, supporting an Intel Pentium III@ 750 MHz CPU with 128 MB of RAM, 4 RS-232 serial ports, 1 CF card slot and Ethernet at 100 Mbps, and connected to three PC-104 modules supporting digital I/O, analog input and analog output respectively.

Currently, the system computing platform is being upgraded in order to allow the handling of image acquisition and processing. To this aim, the previous SBC board, expanded with a Leutron Pic Port color frame grabber mounted on its PCI expansion slot, will be devoted to video acquisition and processing, while the control system will run on a Diamond Hercules SBC mounting a 750 MHz low-power processor with 256 MB of RAM, 4 RS-232/RS-485 serial ports, 32 channels 16-bit A/D, 4 channels 12-bit D/A, 40 digital I/O lines, 1 CF card slot, Ethernet at 10/100 Mbits, and connected to one PC-104 module with 8 serial ports software configurable as RS-232/RS-422/RS-485 up to 115.2 kbps and 8 digital I/O lines.

Communications with the remote control and supervision station are guaranteed by a radio wireless LAN at 1.9 Mbps, supporting robot telemetry, operator commands, and video image transmission.

The human operator station is constituted by a laptop computer, running a Human Computer Interface, and the power supply system, which integrates a couple of solar panels (32 W at 12V) and one lead battery (100 Ah at 12 V).

3. Software and Control Architecture

The basic requirement in designing and implementing the robotic vessel software architecture was to define a common infrastructure for managing an embedded system supporting real-time thread scheduling and communications, and the levels of an intelligent control architecture, including asynchronous modules implementing potentially unbounded search algorithms.

Recent advances in computing power of COTS boards and in performance of free software, e.g. GNU/Linux, which allow accurate 10-Hertz controller updates without requiring special purpose real-time kernels such as VxWorks, RTLinux or RTKernel (see, for instance, the Jason II control system [Whitcomb et al., 2003]) motivated the choice of the SBC, PC-104 and GNU/Linux based hardware and software platform described in the previous section.

Summarizing the architecture consists of:

- an embedded real-time infrastructure constituted by a real-time custom scheduler, which guarantees an effective scheduling of the control application threads according to the required priorities and policies, integrated with a set of utilities for thread creation and data access synchronization and communication;
- an intelligent control system, constituted by the device drivers, execution level, running navigation, guidance and control tasks, and execution control level, guaranteeing data consistency and automatic reconfiguration of the control system (Caccia et al., 2005c). Recently an event handler module, able to generate events on the basis of the temporal evolution of NGC and plant variables, and a basic path-planner, computing the vehicle path through user defined way-points and predefined line shapes, have been integrated in the control system architecture in view of the development of a mission control system.

In the following the basic concepts of the developed software and control architecture will be discussed, referring to other specific papers for a detailed description of the adopted methodologies and algorithms.

3.1 GNU/Linux-based Embedded Real-time Operating System

As mentioned above, the implementation of the Jason II control system demonstrated that a standard GNU/Linux distribution is sufficient for real-time requirements of unmanned marine vehicles. The need to develop a platform able to support generic applications requiring higher sampling and scheduling rates, such as, for instance, the introduction of an electro-mechanical robotic arm on-board an ROV, motivated further studies on the performance limits of standard GNU/Linux for embedded real-time systems. In particular, the research, partially funded by a SME involved in the development of marking machines in the field of iron and steel factories, focused on the development of an embedded real-time platform based on standard GNU/Linux for industrial automation and robotics. In this framework the Charlie USV architecture has been a simple test case for preliminary evaluation of the results.

A detailed description of the developed methodology can be found in Bruzzone et al. (2006), while for more information about the Charlie USV application the reader can refer to Bruzzone and Caccia (2005).

Here it is sufficient to remind how a native fair Unix-like system such as GNU/Linux could be used for real-time applications with strict time requirements beginning from the insertion as a standard kernel configuration option of the so-called preemption and low-latency patches. From the kernel release 2.5.4-pre6 this led to maximum latency[1] and jitter [2] of the order of a few tens of milliseconds also when particularly demanding activities, which are usually avoided by typical real-time embedded system programming techniques, are performed (e.g. video output, dynamic allocation of large amount of memory, process fork, etc.). Thus it was possible to implement a real-time custom scheduler for real-time threads, both synchronous and asynchronous, while non-real-time ones continue to be scheduled under the default universal time-sharing Linux scheduler policy. The custom scheduler is based on interrupts generated by a suitable timer, such as the Real Time Clock (RTC)

1. latency: delay between the occurrence of an interrupt and the running of its service thread
2. jitter: variations of timing of periodic events

available on common PC boards, which can generate signals from 2 Hz to 8192 Hz and be read as a conventional blocking device. It is worth noting that the proposed solution allows scheduling all the threads in user space without requiring any modification of the kernel.

When the first release of the embedded real-time system was developed in 2005, it was based on Slackware 10.1 GNU/Linux distribution with a standard Linux 2.6.9 kernel. Threads and synchronization semaphores were implemented using the LinuxThreads library, an implementation of the Posix 1003.1c thread package for Linux. This required the implementation of message queues for inter-thread communications in addition to a set of C++ classes encapsulating both non-real-time and real-time threads blocking on asynchronous events, periodically executed or waiting for timed signals.

The dramatic improvements in size and performance of solid-state memories, such as compact flash (CF) memory cards and RAM, have made less critical the construction of a customized embedded GNU/Linux system. The only warning regards the need of loading and mounting the operating system and user applications in RAM from CF during the start-up sequence in order to avoid any latency due to slow accesses to hard or CF disks.

Results have been fully satisfactory. With the custom scheduler running at 1024 Hz, and the execution level scheduled at 8 Hz, a maximum jitter error lower than 250 µs has been measured in more than 15 hours of at field trials and extended laboratory tests (about two weeks of running connected to a vehicle hardware-in-the-loop simulator).

Moreover, the proposed platform allowed an easy implementation of control system architectures for applications with threads having different real-time requirements, as, for instance, the Charlie and Romeo UMVs and the Green Project's Hammer marking machine (Bruzzone et al., 2006).

In addition to the classical advantages given by the use of a standard, worldwide spread and open source operating system, currently the main benefit of a GNU/Linux based platform is given by the easiness of integration in the developed platform of the continuous dramatic improvements in real-time performances of standard GNU/Linux. This is, for instance, the case of the NPTL (Native POSIX Thread Libraries) libraries, which, thanks to the introduction of futexes (Fast Userspace muTexes) as a building block for fast userspace locking and semaphores, are real-time now (kernel version 2.6.21).

The drawbacks represented by the need of a dedicated CPU for hard disk data logging and other warnings in programming techniques are not a real constraint since they are usually adopted by programmers of embedded real-time systems, and the physical separation of the telemetry and data logging system from the control system increases system reliability.

3.2 Control Architecture

The main novelty in the adopted control architecture has been the full integration of a Petri net based execution control level, as an interface between the synchronous execution of motion estimation and control modules and the asynchronous discrete event systems handling mission control and supervision. This, together with the development of suitable tools for describing the NGC system architecture, dramatically facilitated the upgrade of the control system introducing new algorithms and tasks.

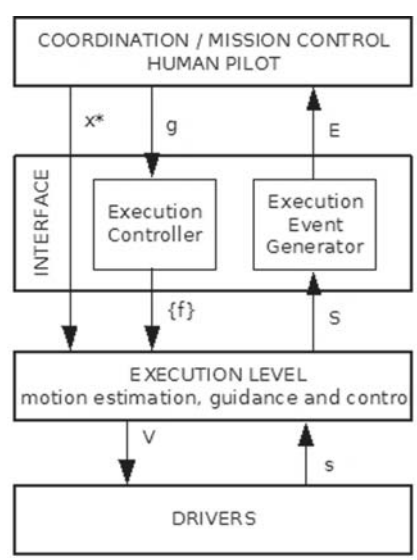
The idea of adding to the conventional intelligent control architecture a layer devoted to guarantee the consistency of the execution level data in the presence of task conflicts and dependencies, bridging the gap between the world of control systems and the one of artificial intelligence, was introduced in the work carried out by CNRS-LAAS on architectures for autonomous mobile robots (Alami et al., 1998), and then applied in the field of marine robotics by CNR-IAN (now CNR-ISSIA) to the Romeo ROV starting from the new century (Caccia et al., 2005c).

According to the general scheme shown in Figure 3, the Execution Event Generator is in charge of monitoring the task state and performances *S* to generate suitable events *E*, signaling particular interactions of the robot with the operating environment and the state of motion estimation and control tasks. On this basis, the mission control level (or human pilot) dynamically schedules the motion control and estimation tasks in order to accomplish the desired mission. The execution control module checks if the command *g* leading the specified Execution Level tasks in the desired state, i.e. running, idle or init, can be executed. In the case that this leads to a forbidden state, the execution control determines a suitable set of task activation/deactivation commands {f} leading consistently to the desired state.

Checks on the admissibility of the user commands were originally inserted in the Romeo ROV software as a set of "*if ... the ... else ...*" clauses, but this solution proved to be unable to manage the alternative execution of more than five-six NGC tasks. Thus, the development of a module able to handle the switching, during the same mission, among different algorithms and navigation, guidance and control schemes was required by the family of UMVs developed by CNR-IAN/ISSIA both for research and operational needs. As a consequence research focused on the definition of a suitable description of the NGC system structure for automatically handling of task conflicts and dependencies at run time. The result, as thoroughly discussed in Caccia et al. (2005c), was the definition of the so-called Task-Variable Graph (TVG), i.e. a diagram expressing the I/O relationships and constraints between the estimate and control variables and the NGC tasks. The constraints on the I/O variables express the rules stating the fundamental properties of the desired control architecture: data consistency in the presence of task conflicts, correct

FIGURE 3

Charlie control architecture.



task dependencies according to the principles of a hierarchical architecture and "from bottom upward activation", i.e. the tasks closer to the hardware are activated before than the other ones. In particular, the analysis carried out revealed that basic relationships, i.e. mutual exclusion and dependency, between the NGC tasks are embedded by the topological structure of the execution level, i.e. by the TVG, thus allowing an automatic management of the corresponding constraints. To this aim, research carried out at the University of Notre Dame on supervisory control of Petri nets using place invariants (Yamalidou et al., 1996; Moody et al., 2000) allowed to treat the problem of TVG representation and online reconfiguration in an elegant and practical way, since the translation of the above-mentioned basic rules on the structure of the control architecture to constraints on a Petri net representing the execution level as a discrete event system is quite easy and can be performed automatically. In particular, as discussed in Caccia et al. (2005b), the implementation of two distinct Petri nets, representing the estimate and control tasks respectively, dramatically increase system performances at the cost of reduced communications between the two modules.

On this basis, the specification in suitable configuration files of the estimate and control variable, NGC algorithms, and relationships between their instances and the variables as well as the definition of templates for C++ algorithm implementation and of a suitable application for the off-line generation of the controlling Petri nets, enables the reduction of the control architecture basically to a synchronous thread executing the NGC tasks and an asynchronous one controlling the automatically generated Petri net implementing the execution control level.

The resulting control system architecture, allowing the run of different NGC schemes, e.g. dual-loop and single loop heading control, during the same exit at sea and the easy introduction of new NGC modules, increased the system capability of handling different operating conditions and executing different missions, when different configurations of active NGC tasks are required, and the possibilities of cooperation with other research groups, facilitating the access to CNR-ISSIA marine robotic platforms.

The interface between the mission control/human operator and the execution level has been completed by the design and implementation of a module, the Execution Event Generator, able to monitor the continuous time evolution of NGC variables and to generate asynchronous events according to the requests of the mission controller, for instance, signaling when time out expires or a variable gets close enough to a specified reference, both a constant or another variable.

In addition, after the experience in the implementation of advanced path-following techniques gained in the cooperation with CNRS-LIRMM (see section 5.3), a path-planner has also been introduced in the architecture, generating suitable sequences of 2-D points according to the user requirements, e.g. splines of n-th order between sequences of way-points, straight lines, circles, and so on.

The upgraded architecture, i.e. with the integration of the Execution Event Generator and Path Planner, has only been tested in simulation, while field trials are foreseen for summer 2007.

4. Modeling, Identification and Navigation, Guidance and Control System

4.1 Modeling and Identification

The general methodology developed and tested for modeling and identifying of the Romeo ROV (Caccia et al., 2000b) has been applied to the Charlie USV in order to define a *practical* model and the corresponding identification procedure, where *practical* stands for consistent, from the point of view of the degree of accuracy, with the quality in terms of noise and sampling rate of the measurements provided by the sensors commonly available on-board a small and relatively low-cost vehicle. As discussed in Caccia et al. (2000b), in the case of poor sensor measurements, special attention has to be paid to the design of the experiments, i.e. of the maneuvers performed by the vehicle, in order to guarantee the observability of the hydrodynamic model parameters: in particular, steady-state maneuvers have been designed for the identification of hydrodynamic drag on the basis of on-board compass and GPS data, while conventional zig-zag maneuvers have been executed for a rough estimate of the yaw moment of inertia, including its added component. The nonlinear model of Blanke for speed and steering equations (Fossen, 1994) integrated with the rudder model of Lewis (Perez and Blanke, 2002), such that the total resultant force is nearly normal to the center plane of the rudder, has been assumed and then simplified on the basis of the experimental data. For a detailed description of the experiments carried out and analysis of the collected data the reader can refer to Caccia et al. (2006). Here it is worth noting that for a small catamaran-shaped USV equipped with GPS and compass:

a) the sway velocity with respect to the water is negligible;
b) the coupled effects of the vehicle speed with the propeller revolution rate and rudder angle are not observable

Thus, the resulting practical model consists of *speed equation*

$$m_u \dot{u}_r = k_{u^2} u_r^2 + k_{n^2\delta^2} n^2 \delta^2 + n^2$$

and *yaw steering equation*

$$I_r \dot{r} = k_{r|r|} r|r| + n^2 \delta$$

where n is the propeller revolution rate, proportional to the reference thruster voltage, d is the rudder angle, u_r and r are the vehicle surge speed with respect to the water and yaw rate, m_u and I_r represent the normalized surge and yaw inertia, and a quadratic drag has been verified both for surge and yaw motion as well as rudder drag in surge motion, which as a consequence of observation b is function of the propeller revolution rate n and not of the surge speed u_r.

The yaw steering equation holds only for n sufficiently far from zero because, due to the observation b, the steering torque has been identified as function of the propeller revolution rate n instead of the advance speed u_r, thus neglecting the rudder action when the vehicle is still moving without rotating the propellers.

As shown in the following, the above-presented model allows a precise and smooth estimate of the vehicle linear and angular position and speed as well as satisfactorily performances in guidance and control.

Unlike in the case of the Romeo ROV, where the execution of maneuvers for the vehicle dynamics identification based on on-board sensors is quite easy, at least for the most critical heave and heading motion, the main drawback for the on-board sensor-based identification of a small USV using only GPS and compass is given by the difficulties in finding suitable weather conditions, i.e. negligible sea current and wind, for test execution. This increased the interest for the transfer of well-known techniques for auto-tuning PID, which require a set of maneuvers very easy to be executed, from the industrial automation area to the field of marine robotics (see section 5.4).

4.2 Model-based Motion Estimation

As far as motion estimation is concerned, a model-based Kalman filter has been applied, providing good results in terms of precision and smoothness of the estimated linear and angular position speed. In particular, the Charlie USV experimental results demonstrated, as well as the experience gained with the Romeo ROV, that the virtual acceleration measurements provided by an accurate model of the vehicle dynamics allow a smooth estimate, and not particularly delayed, of the vehicle speeds, which can be used by motion controllers with relatively high gains. For a detailed discussion of the Charlie model-based motion estimation the reader can refer to Caccia et al. (2007).

In the following some anomalies verified in the GPS measurements are reported as well as the adopted countermeasures. Indeed, the Ashtech GG24C provides piecewise continuous latitude and longitude signals, presenting steps of the type shown in the left part of Figure 4, where the blue lines in the first and third graph from top represent the raw GPS x and y measurements. On the other hand, excepting the discontinuity points, the position signal is quite smooth with a standard deviation of the noise of about 0.17 m. These results seem to confirm the fact, discussed in Ochieng et al. (2002), that although the artificial degradation of the signal through the process of selective availability was removed in 2000, the accuracy of GPS, when used in instantaneous stand-alone mode, remains of the order of 10-20 m. Thus, the Kalman filter for linear motion estimation has been upgraded with an on-line detector of signal discontinuities.

Results are shown in the right plot of Figure 4, where the estimated x-y path is represented by the red line in the right picture. As shown in the second and fourth graphs from top in the left column, the estimated linear speeds along the x and y axis are continuous.

4.3 Guidance and Control

In the field of USV guidance and control, the main lesson gained by the operating experience with the Charlie USV is that many practical applications do not require the design and implementation of advanced nonlinear controllers. For instance, basic line-of-sight guidance and PD heading control were sufficient for controlling the vehicle motion when collecting water samples from the sea surface microlayer and immediate sub-surface during the XIX Italian Expedition to Antarctica in 2004 (see section 5.1).

Moreover, the possibility of implementing with satisfactorily performances a dual nested loop architecture decoupling the management of dynamics and kinematics, i.e. speed and position control respectively, using only linear and angular position sensors, i.e. GPS and compass, has been demonstrated. This guidance and control structure facilitates the design of guidance task functions at the kinematic level, i.e. position controllers generating reference speed signals, implementing different vehicle maneuvers from basic auto-heading to straight line and path following, which can be easily managed by the control system as a consequence of the introduction of the Execution Control level.

Velocity control relies on a model-based gain-scheduling technique (Khalil, 1996) already satisfactorily adopted for controlling the linear and angular speed of the Romeo ROV. For a deeper discussion of the implemented gain-scheduling speed controller and of the application of this control scheme to

FIGURE 4

Measured (blue) and estimated (red) vehicle position. In the left graph, the estimated vehicle speed in the earth-fixed reference frame are plotted too (second and fourth plot from the top).

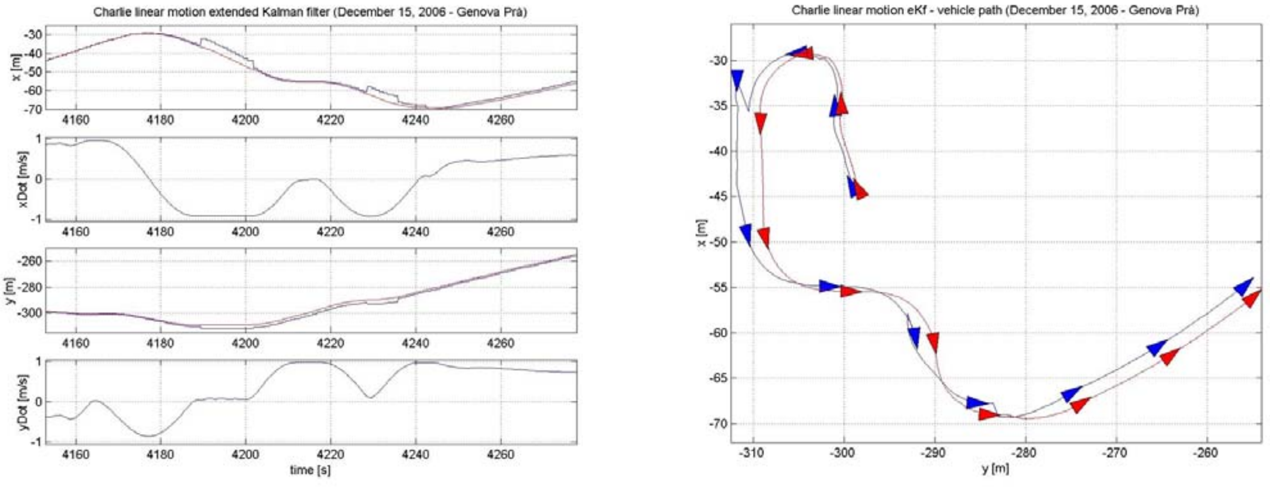
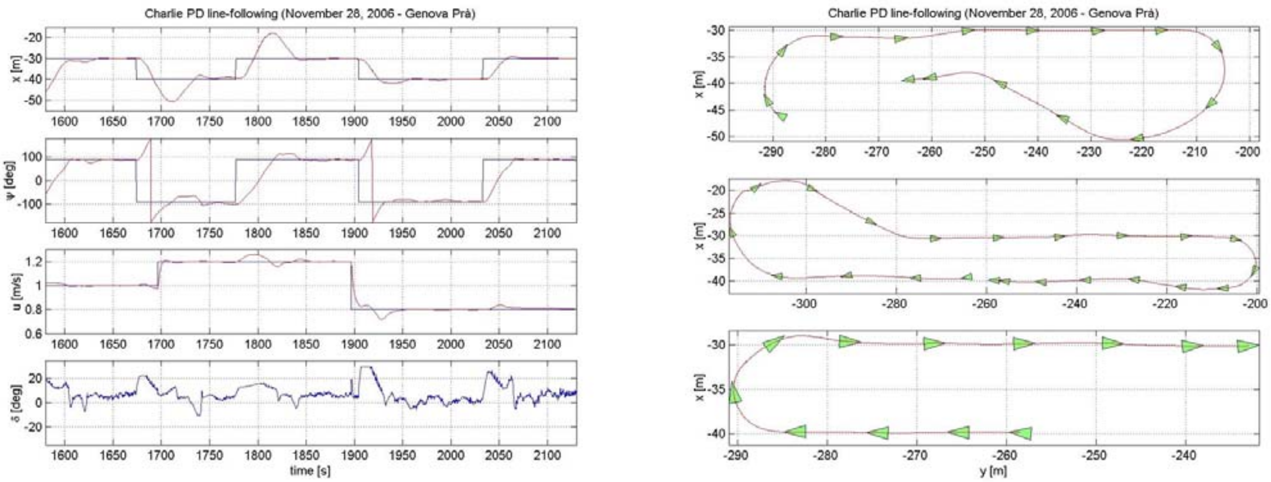


FIGURE 5

PD straight line-following. Left figure, from top to bottom: reference (blue) and estimated (red) straight line x coordinate; reference line orientation (blue) and estimated vehicle heading (red); reference (blue) and estimated (red) vehicle surge speed; reference rudder angle. Right figure: USV path during straight line following trials.



marine vehicles the reader can refer to Caccia et al. (2000c). According to this approach, a simple PI controller, generating the reference vehicle yaw rate, has been designed and implemented for performing auto-heading. Thanks to the physical properties of the system, simple position controllers can be defined for executing more complex tasks too. This is, for instance, the case of straight line following, where sea current and wind disturbances are naturally compensated by the presence of a physical integrator between the vehicle yaw rate and the angle of attack between the desired line orientation and the vehicle heading. Thus, a simple PD straight line-following controller generating the reference yaw rate is sufficient to zero the static error, as shown by experimental results reported in Figure 5. During the tests, carried out in the Genova Prà harbor, a sheet of restricted water usually swept by a 20-30 knots wind, the vessel was commanded to follow lines parallel to the y axis, alternatively eastward, i.e. line orientation equal to 90 degrees, with x equal to -30 m and westward changing its x coordinate to -40 m. The line was correctly tracked with smooth heading variations and rudder action.

For a deeper discussion of the guidance and control algorithms implemented and tested on the Charlie USV the reader can refer to (Caccia et al., 2007).

5. Exploitation

5.1 Sea Surface Autonomous Modular unit (SESAMO) project

The first version of the Charlie catamaran was basically designed to satisfy the operational requirement of sampling the sea surface microlayer and immediate sub-surface water for the study of sea-air interactions in the Ross Sea, Antarctica, in cooperation with the scientific end-users of Consiglio Nazionale delle Ricerche – Institute for the Dynamics of Environmental Systems, Venice. In the framework of the Sea Surface Autonomous Modular unit (SESAMO) project, funded by the Italian National Program of Research in Antarctica (PNRA), the vessel was equipped with a scientific payload consisting of a rotating glass drum for microlayer collection, a seawater intake for immediate sub-surface water sampling, and a number of teflon-membrane pumps and three-way valves in order to distribute the sampled water in suitable stocking buckets for organic and inorganic samples. The system

FIGURE 6

Charlie launching in Genova harbor, October 22nd 2003.

could stock up to 20 l of organic samples and 10 l of inorganic ones both for microlayer and immediate sub-surface, for an overall load of about 60 Kg. In order to maintain about constant the vessel weight during sampling, a couple of ballast water tanks were positioned in the hulls to be emptied in the course of the mission. The rotating glass drum was 0.33 m in diameter and 0.50 m in length, and, actioned by an electrical brushless thruster, could rotate between 4 and 10 rpm.

After its launch and basic tests in the Genova harbor in October 2003 (see Figure 6), the Charlie catamaran was exploited in the framework of the PNRA-SESAMO project in the proximity of the M. Zucchelli station of Terra Nova Bay, Ross Sea, during the XIX Italian Expedition to Antarctica in January-February 2004. The Charlie USV was deployed and recovered by the Malippo, a 16m support vessel hosting the human operator station.

Although the vehicle performed only proportional-derivative heading control and line-of-sight guidance, which relied on a steering mechanism based on the differential propeller revolution rate, these basic guidance and control capabilities were sufficient to satisfactorily execute the required task. Compatibly with weather and sea conditions, the robotized catamaran executed six missions (two for communications, navigation, guidance and control tests and four for water sampling) working for more than 20 hours and collecting about 95 l both of microlayer and immediate sub-surface water samples. A view of the Charlie USV collecting samples among the Antarctic ices is given in Figure 7.

A detailed discussion of system requirements for the study of sea-air interface, guidance and control system, and Antarctic exploitation can be found in Caccia et al. (2005a).

5.2 Coastal and Harbor Underwater Anti-intrusion System

In the period 2005-2007, the Charlie USV is being exploited in the project "Coastal and harbor underwater anti-intrusion system" funded by PRAI-FESR Regione Liguria. The project studies the feasibility of an underwater anti-intrusion system for very shallow water through the experimental evaluation of some key technologies. In this context, the Charlie vessel will support basic experiments for evaluating the performance of high speed acoustic modems between moving stations in harbor waters. As far as USVs are concerned, the main goal of the project is to evaluate the possibility of using this class of vehicles for harbor bottom-mapping in the presence of usual traffic. This goal involves the development of high precision navigation, guidance and control modules and user friendly and reliable mission control system, as well as the evaluation in terms of performance and system integration of sensors for detecting obstacles, e.g. ships, boats and peers, such as radars and optical cameras. In particular, research on USV modeling, identification, navigation, guidance and control, summarized in section 4, constitutes a part of the activity scheduled in this project. Figure 8 shows the vehicle during NGC trials.

5.3 Sensor-based Guidance and Control of Autonomous Marine Vehicles: Path-following and Obstacle Avoidance

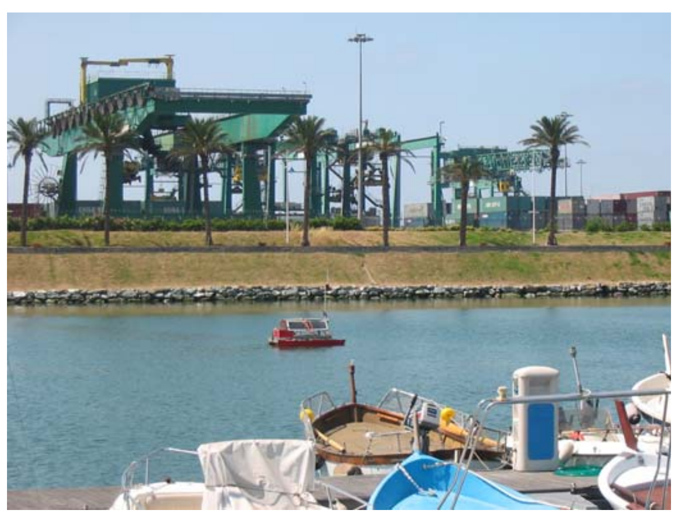
In the period 2006-2007, the Charlie USV is being exploited in the framework of a bilateral cooperation between Italian CNR and French CNRS. The project "Sensor-based guidance and control of autonomous marine vehicles: path-following and obstacle avoidance", in cooperation with the Laboratoire d'Informatique, de Robotique et de Microélectronique de Montpellier, focuses on the development of a sensor-based guidance system for path-following and obstacle avoidance able to be integrated both on the Charlie USV and Taipan AUVs (Paim et al., 2005). The first goal is the integration of path-following and obstacle avoidance modules, developed by CNRS-LIRMM, with the CNR-ISSIA USV and its demonstration at field with the catamaran executing user-defined paths. In this context, nonlinear path-following techniques based on the concept of *following the rabbit*, i.e. controlling also the motion of a target vehicle on the path, using it as an extra degree of

FIGURE 7

Charlie USV sampling the sea surface microlayer in Antarctica (January 31st, 2004).

FIGURE 8

Charlie USV during NGC trials in the Genoa Prà harbor, June 14th 2006.



freedom for the guidance algorithm (Lapierre et al., 2003), have been integrated in the Charlie control system and tested at sea in the Genova Prà harbor in December 2006. In particular, two different configurations have been considered: i) the guidance algorithm has been adapted to generate yaw rate and surge references for standard Charlie speed controllers; ii) a theoretical-based backstepping control design methodology has been adopted to directly compute the reference rudder angle within the path-following algorithms. Experiments were carried out with satisfactory results in the presence of strong wind and sea current. It is worth noting that the above-mentioned tests were, to the authors' knowledge, one of the first experimental validations of the backstepping nonlinear control design method for marine vehicles, which is theoretically able to guarantee stability globally and not only locally, with the assumption of neglecting measurement noise and model uncertainties.

5.4 Auto-tuning Controllers

The Charlie USV is also at the center of a cooperation based on mutual interest and good will for the benefit of common research between CNR-ISSIA and the Control Engineering Group of the Department of Control and Computer Engineering in Automation of the Faculty of Electrical Engineering and Computing, Zagreb, Croatia. The main goal is to develop automated auto-tuning procedures, similar to the ones that are so common in the process industry with PID controllers, that are easy to use and do not need expert knowledge from the USV operator. This capability is of great importance for autonomous marine vehicles especially if their dynamics changes due to various reasons. Tests at sea, carried out in the Genova Prà harbour with the Charlie USV in very bad environmental conditions (strong wind and sea current) at mid December 2006, have provided satisfactory results.

5.5 Support for the Development of a Multi-purpose USV

In 2007-2008 the Charlie USV will support the development, testing of sub-systems, and performance evaluation through comparative trials, of a multi-purpose prototype USV that CNR-ISSIA is going to design, develop and test in the framework of a feasibility study funded by the Parco Scientifico Tecnologico della Liguria. The new prototype USV will be characterized, according to the specifications of the customer, by an aluminum hull and the capability of deploying and recovering simple instruments for underwater data collection. This project will be accomplished by CNR-ISSIA (project coordinator) in partnership with Gem Elettronica, in charge of the navigation sensor package, radar system, and communication package, Green Project, in charge of the plant automation, and Cantieri Mancini, that will design and build the aluminum vessel.

Concluding Remarks

This paper has introduced the reader to the Charlie project, carried out by the *Autonomous robotic systems and control* group of CNR-ISSIA, Genova, Italy, with the goal of developing autonomous, or loosely remotely controlled, marine surface vehicles able to operate in areas characterized by everyday commercial, fishery and recreation traffic. In the authors' opinion, there are a couple of issues, strictly related between them, separating the current state-of-the-art from the possibility of fully operating USVs in coastal and harbor areas for accomplishing surveying and monitoring tasks: a reliable capability of detecting and avoiding obstacles and legal issues, mainly related to liability in case of collisions, as clearly discussed in the "Issues concerning the rules for the operation of Autonomous Marine Vehicles (AMVs) – A consultation paper," published by the Society for Underwater Technology, and available at *http://sig.sut.org.uk/urg_uris/URG_AMV_paper.pdf*. In particular, the implementation of reliable anti-collision systems, rather than from the development of algorithms for obstacle avoidance according to the *rules of the road*, depends upon the availability of sensors able to detect other vessels and objects at sea, e.g. diver buoys, both in daylight and night conditions and from the capability of interacting with other vessels commanders and human-managed centers for traffic control. Thus, future research should focus on the integration of different sensors, such as radar and standard daylight, FLIR and LGI CCD vision systems, for obstacle detection and the development of new paradigms for the interactions between human-operated systems and (semi-)autonomous intelligent vehicles.

Acknowledgments

This work has been partially funded by the Italian National Program of Research in Antarctica (PNRA), field of application Technology, project SESAMO, by PRAI-FESR Regione Liguria "Coastal and harbor underwater anti-intrusion system" project, by the CNR-CNRS bilateral agreement under the project "Sensor-based guidance and control of autonomous marine vehicles: path-following and obstacle avoidance," and by Regione Liguria - Obiettivo 2 (2000-2006) Sottomisura 1.4B- L. 598/94 art. 11 "Interventi per la ricerca industriale e lo sviluppo precompetitivo" in the project "Embedded real-time platform for industrial automation and robotics."

Special thanks for their kind, enthusiastic and fruitful cooperation are given to Dr. Lionel Lapierre from CNRS-LIRMM, Montpellier, France, and to Professor Zoran Vukic and Dr. Nikola Miskovic from the Faculty of Electrical Engineering and Computing, Zagreb, Croatia.

References

Alami, R., Chatila, R., Fleury, S., Ghallab, M. and Ingrand, F. 1998. An architecture for autonomy. Inter J Robot Res. 17:315–337.

Benjamin, M.R. and Curcio, J. 2004. COLREGS-Based Navigation in Unmanned Marine Vehicles. In: IEEE Proceedings of AUV-2004, Sebasco Harbor, USA.

Bruzzone, G. and Caccia, M. 2005. GNU/Linux-based architecture for embedded real-time marine robotics control systems. In: Proc. of IARP Int. Workshop on Underwater Robotics, pp. 137-144. Genova, Italy.

Bruzzone G., Caccia, M., Bertone, A., and Ravera G. 2006. Standard Linux for embedded real-time manufacturing control systems. In: Proc. of 14th IEEE Mediterranean Conference on Control and Automation, Ancona, Italy

Caccia, M., Bono, R., Bruzzone, G. and Veruggio, G. 2000a. Unmanned Underwater Vehicles for scientific applications and robotics research: the ROMEO Project. Mar Technol Soc J. 24(2):3-17.

Caccia, M., Indiveri, G. and Veruggio, G. 2000b. Modelling and identification of open-frame variable configuration unmanned underwater vehicles. IEEE J Oceanic Eng. 25(2):227–240.

Caccia, M. and Veruggio, G. 2000c. Guidance and control of a reconfigurable unmanned underwater vehicle. Control Eng Pract. 8(1):21-37.

Caccia, M., Bono, R., Bruzzone, Ga., Bruzzone, Gi., Spirandelli, E., Veruggio, G., Stortini, A.M. and Capodaglio, G. 2005a. Sampling sea surface with SESAMO. IEEE Robot Autom Mag. 12(3):95-105.

Caccia, M. and Bruzzone, G. 2005b. Execution control of robotic tasks for marine systems. Proc. of IFAC World Congress 2005. Prague, Czech Republic.

Caccia, M., Coletta, P., Bruzzone, G. and Veruggio, G. 2005c. Execution control of robotic tasks: a Petri net-based approach. Control Eng Pract. 13(8):959-971.

Caccia, M., Bruzzone, G., Bono, R. 2006. Modelling and identification of the Charlie2005 ASC. Proc. of 14th IEEE Mediterranean Conference on Control and Automation, Ancona, Italy.

Caccia, M., Bibuli, M., Bono, R., Bruzzone, G. 2007. Basic navigation, guidance and control of the Charlie2005 ASC. In: Proc. of European Control Conference 2007, Kos, Greece

Cornfield, S., Young, J. 2006. Unmanned surface vehicles - game changing technology for naval operations. In: Advances in unmanned marine vehicles, IEE Control Series, pp. 311-328.

Ebken, J., Bruch, M. and Lum, J. 2005. Applying UGV Technologies to Unmanned Surface Vessel's. In: SPIE Proc. 5804, Unmanned Ground Vehicle Technology VII.

Fossen, T.I. 1994. Guidance and Control of Ocean Vehicles. England: John Wiley and Sons.

Harvey, G.W. 1966. Microlayer collection from the sea surface. A new method and initial results. Limnol Oceanogr. 11:608-613.

Khalil, H.K. 1996. Nonlinear systems. Prentice Hall.

Lapierre, L., Soetanto, D., Pascoal, A. 2003. Nonlinear path following with the applications to the control of autonomous underwater vehicles. In: Proc. of 42nd IEEE Conference on Decision and Control, pp. 1256-1261.

Majohr, J. and Buch, T. 2006. Modelling, simulation and control of an autonomous surface marine vehicle for surveying applications Measuring Dolphin MESSIN. In: Advances in unmanned marine vehicles, IEE Control Series, pp. 329-352.

Manley, J.E., Marsh, A, Cornforth, W. and Wiseman, C. 2000. Evolution of the autonomous surface craft AutoCat. In: Proc. of Oceans'00. 1:403-408.

Moody, J. and Antsaklis, P.J. 2000. Petri Net Supervisors for DES with Uncontrollable and Unobservable Transitions. IEEE T Automat Contr. 45(3):462-476.

Ochieng, W. and Sauer, K. 2002. Urban road transport navigation: performance of the global positioning system after selective availability. Transportation Research Part C: emerging technologies. 10:171–187.

Paim, P.K., Jouvencel, B. and Lapierre, L. 2005. A reactive control approach for pipeline inspection with an AUV. Proc. of Oceans 2005. 1:201-206.

Pascoal, A. et al. 2000. Robotic ocean vehicles for marine science applications: the European ASIMOV Project. In: Proc. of Oceans 2000.

Perez, T. and Blanke, M. 2002. Mathematical Ship Modelling for Marine applications. Technical Report. Technical University of Denmark. http://www.iau.dtu.dk/secretary/pdf/TP-MB-shipmod.pdf.

Whitcomb, L., Howland, J.C., Smallwood, D.A., Yoerger, D. and Thiel, T.E. 2003. A new control system for the next generation of US and UK deep submergence oceanographic ROVs. In: Proc. of the 1st IFAC Workshop Guidance and Control of Underwater Vehicles, pp 137–142. Newport, UK.

Xu, T., Chudley, J. and Sutton, R. 2006. Soft computing design of a multi-sensor data fusion system for an unmanned surface vehicle navigation. Proc. of 7th IFAC Conference on Manoeuvring and Control of Marine Craft.

Yamalidou, K., Moody, J., Lemmon, J. and Antsaklis, P.J. 1996. Feedback Control of Petri Nets Based on Place Invariants. Automatica. 32(1):15-28.