Sparus\textsuperscript{AUV} takes the SAUC-E’11 challenge

Enric Galceran, Carlos Becker, Chee Sing Lee, Simone Zandara, Arnau Carrera, Pere Pares, Carles Candela and Marc Carreras

VICOROB-UdG Team
University of Girona, Spain
http://eia.udg.edu/sauce

Executive Summary—In 2006, a pioneer team of students of the University of Girona (the VICOROB-UdG Team) decided to design and develop an Autonomous Underwater Vehicle (AUV), called Ictineu\textsuperscript{AUV}, to face the first Student Autonomous Underwater Challenge - Europe (SAUC-E). After being the winning entry of SAUC-E’06, Ictineu\textsuperscript{AUV} was extensively used as a research platform in the Underwater Robotics Lab of the University of Girona. Four years later, in 2010, a renewed, multidisciplinary team of students took the challenge of building a new AUV aimed to compete again in SAUC-E and at the same time provide a totally different vehicle concept: Sparus\textsuperscript{AUV}. The prototype, designed with simplicity, low weight and modular design in mind, evolved from the initial Computer Aided Design (CAD) model to become a functional AUV in the short period of six months. In this same time period, a completely new software architecture was also developed to operate the vehicle. Sparus\textsuperscript{AUV} took the SAUC-E’10 challenge and achieved to be the winning entry. This year, a renewed VICOROB-UdG Team is facing the SAUC-E challenge again.

I. INTRODUCTION

Student Autonomous Underwater Challenge Europe (SAUC-E) [1] is a Europe-wide competition for students to foster the research and development in underwater technology. In 2010, the competition was organised by NURC, a NATO Research Centre located in La Spezia, Italy and so it is in the present year. The competition is held in a sheltered harbour with salt water and limited visibility, a significant change from previous editions, adding the difficulty of performing in a realistic marine environment. Having participated in the 2006 and 2010 editions, in January 2011, a team of students from the University of Girona (Spain) decided to face again the SAUC-E challenge. As an important difference compared to the other two past entries, this time we are facing SAUC-E with an already functional vehicle, Sparus\textsuperscript{AUV}. However, key hardware modifications have been applied to the vehicle and its software architecture have been redesigned according to acquired experience and implemented from scratch with the aim to take the SAUC-E challenge. Therefore, the VICOROB-UdG Team has been intensively working to make Sparus\textsuperscript{AUV} able to specifically face every task of the competition. This paper describes the Sparus\textsuperscript{AUV} as an entry to the 2011 SAUC-E competition. The paper is organized as follows. The mechanical, hardware and software design are explained in sections II to IV. Section V describes all the software components specifically designed for the SAUC-E’11 mission, including the navigation and localisation systems as well as all the adopted solutions for every mission task. Section VI underlines the hardware and software innovations. Finally, sections VII and VIII present the financial summary and the risk assessment respectively before concluding the paper in section IX.

II. MECHANICAL DESIGN

<table>
<thead>
<tr>
<th>TABLE I: Sparus\textsuperscript{AUV} features</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Dimensions</strong></td>
</tr>
<tr>
<td><strong>Dry Weight</strong></td>
</tr>
<tr>
<td><strong>Umbilical</strong></td>
</tr>
<tr>
<td><strong>DOFs</strong></td>
</tr>
<tr>
<td><strong>Computer module</strong></td>
</tr>
<tr>
<td><strong>O.S.</strong></td>
</tr>
<tr>
<td><strong>Sensors</strong></td>
</tr>
</tbody>
</table>

After four years of research work with the Ictineu\textsuperscript{AUV}, the need for a new platform was frequently brought to the table. The participation in the SAUC-E competition was the impetus for this idea to become reality. Although we needed a vehicle to fulfill the competition requirements, we decided that it should
be built in a way that would allow exploration of new research lines in our university.

Based on our previous experience and feature needs, we set the following basic requirements:

- Torpedo shaped vehicle
- Total dry weight $< 35$ Kg
- Power autonomy $> 2$ hours
- Hovering capability
- Compact, small size
- Minimum downtime between missions
- Simplicity for easy maintenance
- Construction with easy to find materials
- Robustness

With $1.22$ m length, $0.23$ m diameter and $30$ Kg Sparus$^{\text{AUV}}$ can be even manipulated by a single person (Fig. 1). The largest parts of the vehicle structure are the two aluminium pressure housings. One contains the batteries and monitoring electronics and the other contains the PC, hard drive, electronic boards and the MRU. Having the batteries in a separate housing increases the weight, length and expense of the vehicle but on the other hand it minimizes the between missions downtime by allowing battery packs to be quickly interchanged. Also, potentially explosive gases that can build up from the batteries do not interfere with sparking and high temperature electronics. The aluminum easily transmits the internal heat to the environment, it is easy to machine and is strong enough to withstand the sea pressure. We wanted the housings to be rated for $100$ m depth and the calculations indicated a cylinder thickness of $5$ mm and an endcaps thickness of $1$ cm. In order to eliminate corrosions due to sea water and different metals on the skin is covering the AUV. The RC servomotors used an RC servomotor which we needed to encapsulate by- [311x133] making them waterproof. They are mounted on the top of the vehicle to emerge and get GPS and Wireless data when the vehicle is at surface. It also has the option to be detached and float in the surface with a buoy, keeping the connection with the AUV via a $5$ meter extension cable. Finally, to reduce the water drag, a two part ABS made in a specialized machine shop. The structure is built in a way that would allow exploration of new research lines in our university.

The main battery switch is a magnetic switch, which is used to power an electronic board contained in the battery housing that controls a solid state relay. When the voltage of the batteries is above a certain threshold, the relay is switched on. An external wireless access point and a GPS unit have been covered with resin to make them water proof. They are mounted on the top of the vehicle to emerge and get GPS and Wireless data when the vehicle is at surface. It also has the option to be detached and float in the surface with a buoy, keeping the connection with the AUV via a 5 meter extension cable. Finally, to reduce the water drag, a two part ABS skin is covering the AUV.

Table I shows the main features of Sparus$^{\text{AUV}}$.

### A. Fins system development for pitch control

As an important innovation, this year we decided to equip Sparus$^{\text{AUV}}$ with the ability to control its pitch DOF. To do so, a system composed of two fins actuated by two servomechanisms has been designed and developed.

1) Fins design: We choose the fins shape to follow the standard NACA 0015 profile, which is a symmetric profile. It applies a lift force $F$ in the pitch DOF given by:

$$F = C A \sin(\alpha) 0.5 \rho v^2$$

where:

- $F$: lift force on the fin [Kg]
- $C$: drag coefficient
- $A$: fin area [m$^2$]
- $v$: vehicle velocity [m/s]
- $\rho$: water density [Kg/m$^3$]
- $\alpha$: fin angle [deg]

2) Fins actuator design: In order to actuate the fins, we need an motor that allows for position control. We used an RC servomotor, which we needed to encapsulate in order to make it waterproof. The RC servomotors used
Fig. 1: Sparus\textsuperscript{AUV} mechanical design. Top view shows the robot with the external skin. Middle view shows the distribution of the foam. Down view shows internal devices.

were SC-1251MG from Savox. A PCB was designed and built in order to generate the appropriate control signals and to make Sparus\textsuperscript{AUV}'s computer able to control the fins system. The PCB is interfaced through a USB bus. Fig. 2 shows the CAD model with the designed fins system.

3) Fins system development process: The fins have been hand made by using an also hand made mold and resin. Fig. 3 shows an advanced step of the development process.

III. HARDWARE DESIGN

A. Computer Module

The on board embedded computer (Fig.4) has been chosen as a trade off between processing power, size and power consumption. The ADL945HD board together with the U2500 processor at 1.2 Ghz provides the processing power of a Core Duo architecture together with the Ultra Low Voltage (ULV) consumption and the 3.5" small form factor. Moreover, a high number of ports (8 USB’s, 4 RS-232, 2 Ethernet ports, etc.) are directly available on the board to connect the Sparus\textsuperscript{AUV} wide set of sensors. To ensure the robustness of this computer module, a combination of a heat spreader and two heat pipes in contact with the housing are used to maintain the right temperature inside the cylinder and a solid state disk is used to reduce the risk of mechanical failures.

B. Power module

The vehicle power module (Fig. 5 consists of one battery pack at 25.2V. This battery gives the power to all the systems: computer, electronics, sensors, thrusters and fins. This battery pack has 13Ah of capacity which allows an autonomy of 3 hours. The pack is composed by
NiMH battery cells (F type, 1.2V, 13Ah). Hence, a total of 21 cells are used for building the pack. The battery pack is connected to the computer housing where a DC-DC converter (PC104 format) generates +5V, +12V and +24V for powering the computer, the electronics, the sensors and the fins. On the other hand, the thrusters are directly connected to the battery pack. The housing containing the batteries can be easily removed from Sparus\textsuperscript{AUV} in a matter of minutes, allowing to quickly swap it for another housing with recharged batteries. This housing contains also an electronic board which measures the voltage and current from both battery packs and also some internal measurements (temperature, pressure, humidity) for safety purposes. It contains also the main battery switch, which is magnetic and can be operated from the top part of the vehicle. The switch is connected to the electronic board, and this one activates a solid state relay if the battery voltage is above 23.5V. The switch has some LEDs to monitor the state of the battery pack, which blink according to the remaining energy. The battery housing does not need to be opened in order to recharge the batteries. External power can be supplied through the magnetic switch Subconn\textsuperscript{R} connector and a small screw is provided to allow ventilation during the charging process. Additionally, an umbilical cable can be connected to the vehicle’s computer housing for ethernet access and external power. Two diodes are used to mix the battery energy and the external energy.

\begin{figure}[h]
\centering
\includegraphics[width=0.4\textwidth]{sparus-auv-battery-housing.png}
\caption{Sparus\textsuperscript{AUV} battery housing.}
\end{figure}

\textbf{C. Sensor suite}

The vehicle is equipped with a complete sensor suite composed by two color video cameras (forward-looking and down-looking), an MTi MRU from XSens Technologies, a Micron imaging sonar from Tritech, an echosounder, a pressure sensor, a Doppler Velocity Log (DVL) from LinkQuest which also includes a compass/tilt sensor and several hydrophones for the acoustic pinger detection. Additional temperature, voltage and pressure sensors and water leak detectors are installed into the pressure vessels for safety purposes.

\textbf{D. Actuators}

In addition to the three thrusters and two fins previously introduced, the vehicle is also equipped for SAUC-E with an actuator to cut a nylon fishing line and free the mid-water target (See mission description in SectionV). This actuator is a servo-based mechanism connected to a domestic pliers which open and close sequentially when and internal electronic board switches on a relay. In order to help the line to reach the small pliers entrance, some aluminum bars are used to guide the line up there.

\textbf{IV. SOFTWARE DESIGN}

A control architecture is a set of software components interacting with the robot’s hardware and to each other hereby guaranteeing the AUV functionality. On the other hand, a control architecture typically uses some underlying software framework to allow communication among its components, which potentially may be in different computers on a network.

The control architecture operating Sparus\textsuperscript{AUV} consist of three main modules:

- Vehicle interface module: consisting of a set of device drivers allowing to gather information from sensors and to send commands to the actuators.
- Perception module: this module’s components take the data gathered from sensors in the vehicle interface module and process it to extract more complex and useful information from it. Examples of components in the perception module might be a mid-water target detector or a localisation system.
- Control module: a set of components that send commands to the robot actuators (in the vehicle interface module) according to the information obtained by the perception module. In this module there are a specific kind of components called behaviors. Those components generate responses for each DOF of the vehicle in order to reach a specific goal. Then these responses are processed by other components (which depend on the control approach used) to end up sending commands to the thrusters in order to reach the behaviors’ goals. A behavior component might be the \textit{AvoidObstacles} behavior or the \textit{GoToPoint} behavior.
An external software component, called the Mission Control System (MCS), orchestrates the control architecture components sending them actions according to events received from the control architecture. For instance, the MCS might send the action *StartWallFollowing* on the arrival of the event *WallDetected*.

In the next subsection, the control architecture used on Sparus$^{[AUV]}$ for SAUC-E’11 will be discussed. After that, on the next subsection, its implementation using the Robot Operating System (ROS) platform will be explained.

A. The COLA2 control architecture design

In the 2006 and 2010 SAUC-E editions, the UdG-VICOROB Team used a control architecture called Object-Oriented Control Architecture for Autonomy (O2CA2). For this year’s edition, we developed a new architecture based on O2CA2’s concepts, but redesigned taking into account the experience gathered while operating the former. Hereby we came up with the Component-Oriented Layer-based Architecture for Autonomy (COLA2) (see its block diagram in Fig. 6). Both architectures consist of the three modules discussed above, and differ only in the control module. Next, we will first discuss the COLA2 design and then compare it to O2CA2 to see its innovations and improvements.

The COLA2 architecture is a behavior-based architecture structured in several modules named Vehicle Interface, Perception and Control modules (Fig. 6), where the different software components (drivers, processing units, behaviors,...) interact to each other. Over these three modules there is an MCS orchestrating them.

The following sections will briefly explain the three modules that compose the architecture as well as the Mission Control System.

1) Interface module: This is the only module that contains software objects that interact with the hardware. There are basically two types of components: sensor components responsible for reading data from sensors and actuator components responsible for sending commands to the actuators. Sensor components for Sparus$^{[AUV]}$ include a Doppler Velocity Log (DVL) with an integrated compass, a Micron imaging sonar, a Motion Reference Unit (MRU), two cameras, a depth sensor and an echosounder. There are also components for safety purposes like water leakage detectors and internal temperature, pressure, voltage sensors that allow for the monitoring of the conditions within the pressure vessels. Actuator objects for Sparus$^{[AUV]}$ include the thrusters, the cutter mechanism and the fins control system for actuating the pitch DOF. Another function of this module is to convert all the data to the International System of Units, as well as to reference all the gathered navigation data to a common reference frame located in the mass center of the vehicle.
2) Perception module: The basic components of this module are: the Navigator, the Map-Based Locator and several Target detectors. The Navigator component estimates the position and velocity of the robot merging the data obtained from the navigation sensors DVL, MRU and depth sensor by using an Extended Kalman Filter (EKF). It can also receive absolute corrections from the Map-Based Locator unit. The Map-Based Locator provides an absolute positioning system based on the Micron sonar returns and the voting algorithm that will be described in section V-B. The target detector takes acoustic or visual images gathered by the vehicle interface module and extracts some features from them. Several target detectors can be programmed in order to detect different objects. Section V-D presents the target detectors implemented for the SAUC-E’11 competition in order to follow an underwater irregular structure and detect a mid-water target. The data produced by this module is directly used by some robot behaviors in the control module.

3) Control module: The control module receives inputs from the Vehicle Interface and Perception modules and sends command outputs to the actuators held in the Vehicle Interface Module. In this way, the control module is completely decoupled from the used physical sensors/actuators. This module is composed by a set of behaviors, a pilot and a velocity controller. A behavior is essentially a function that maps an input from the other modules into a pose setpoint. This pose setpoint can either be a point in the world coordinate frame or a twist (a point relative to the vehicle coordinate frame). Each behavior has a particular goal to achieve, for instance the AchieveDepth behavior is considered to reach its goal when the robot is within an interval around the desired depth. To accomplish this, AchieveDepth tells the pilot to go to a certain depth.

The pilot receives pose setpoints from behaviors. Though several behaviors might be active at a time, the pilot only handles one request at a point in time. This is done assigning priorities to each behavior. Then the pilot only handles a request from a behavior if its priority is equal or bigger than the current request’s. The task of the pilot is to generate the appropriate velocity setpoints to fulfill the pose setpoint requests. Then, these velocity setpoints are sent to the velocity controller.

Finally, the velocity controller component takes the current vehicle velocity from the vehicle’s navigation system and by means of a Proportional Integrative and Derivative (PID) controller generates the required setpoints to be sent to the thrusters. In order to allocate the force of each DOF among the thrusters, the pseudo inverse of the thruster configuration matrix is used [3].

4) Mission Control System: Over the three previous modules there is the Mission Control System (MCS) which is responsible for the sequencing of the mission tasks by selecting at each point of the mission the set of behaviors that must be active. We use two different software packages to describe the robot’s missions, depending on the particularities of the task sequence. On one hand, we use URBI Script, an scripting language which allows for easy sequencing and parallelization of tasks and for easy event-driven programming. On the other hand, we use SMACH, a package from the ROS platform that allows for task sequencing by describing the mission as a Python script. SMACH also offers a visualization tool for graphically monitoring the mission execution. Hence, our mission descriptions are either URBI scripts or Python scripts that send the actions to be done to the control architecture according to the events notified by the latter.

5) Comparing COLA2 to the older O2CA2: In O2CA2, a coordinated behavior approach was used. That is, behaviors generated velocity setpoint responses for each DOF and then a component called the coordinator merged these responses according to the behavior priorities. Then, this coordinated velocity response was sent to the velocity controller, which translated it into thruster commands. This approach implied generating a velocity response when programming a behavior, which is not straightforward. For instance, to achieve a certain depth, one needed to generate a velocity response depending on the error (the difference between the current depth and the goal depth), typically using a PID controller. This fact made the behavior development a complex and tedious task.

In the new architecture used this year, COLA2, the control module is designed applying a different paradigm. Instead of the coordinator, there is a component called the pilot. The pilot can handle requests from behaviors in order to navigate to a certain point (either relative to the world frame or to the vehicle frame). Then the pilot sends the appropriate velocity setpoints to the velocity controller, handling only the request from the most prioritary behavior at a time. A major difference with the former approach is that behaviors generate pose responses, not velocity responses. This is more intuitive to the programmer and eases the behavior programming, simplifying the code.
B. The COLA2 control architecture implementation

Implementing a control architecture for a robot is a complex task, as it implies facing a big pile of software components that need to pass information to each other, and that possibly are running in different machines. There are several frameworks, middleware and software libraries addressing the issues that show up when developing robot software. The UdG-VICOROB Team has used different options in the past, depending on the solution used at the Underwater Robotics Lab from the University of Girona at a certain point in time. For SAUC-E’06, a Common Object Request Broker Architecture (CORBA) standard compliant middleware was used for software component communication, but it generated really difficult to maintain code and also slowed down the development process. For SAUC-E’07, we used a own-designed communication framework based in standard XML strings over TCP/IP connections, resulting in a lightweight protocol which allowed to perform the component communication in a much more simple way than other available solutions. Despite it gave us great results and was also used for research tasks in the Underwater Robotics Lab in a satisfactory way, recently we choose to use a more standard and widely-used solution, and we found that ROS (Robot Operating System) fulfilled our needs. ROS is an open source project providing libraries and tools to help software developers create robot applications. Also, a wide community of developers provides new software packages and support. Hence, we decided to move to ROS and take advantage of the off-the-shelf software packages, documentation and on-line support available for ROS developers by implementing the COLA2 architecture using ROS.

V. SAUC-E’11 Mission

The SAUC-E’11 competition takes place in a salt water basin 120m long and 50m wide with a constant depth of 5.5m. The mission to be performed consists in:

• Pass through a validation gate.
• Make a 180 turn, follow an irregular underwater structure while maintaining a 0.5 m stand-off distance from it and pass through a second gate.
• Localize and free a midwater target.
• Perform a wall inspection maintaining a constant distance from it (>2m).
• Perform tracking of a moving Autonomous Surface Vehicle (ASV), either by pinger detection or by looking up with a camera. As will be explained in section V-G, we choose to use the pinger to face this task.
• Surface within a designated surfacing zone which will be marked by means of the ASV, which will be stationary in the competition arena for this task.

In order to fulfill this mission, several software components have been developed. The main ones are described in the following sections:

A. Navigation

The navigation system uses an Extended Kalman Filter (EKF) to estimate the vehicle’s pose and velocity at any given time instance. The EKF state vector comprises the 3D Cartesian position and velocity estimates respectively:

\[ X_k = [x_k, y_k, z_k, u_k, v_k, w_k]^T \] (1)

The position estimates are in the global reference frame, while the velocity estimates are in the vehicle frame. At the mission start, before submerging, the location measurement from the onboard GPS is used to initialize the state vector and covariance matrix. Following standard EKF procedure, the navigator propagates the state estimate through time as a function of the previous state and the control inputs. In this case, the control inputs are accelerations along the vehicle axes:

\[ u_k = [\ddot{u}, \dot{v}, \dot{w}]^T \]

The axial accelerations are derived from the vehicle thrusters’ setpoints, with a linear viscous damping model. In the case of \( \ddot{v} \), only the damping model is applied since the thruster configuration does not allow for actuation in that direction. The following expression is used to obtain the acceleration:

\[ \ddot{u} = \frac{\tau - Du}{m} \]

Where \( \tau \) is the thruster setpoint, \( D \) is a damping coefficient, and \( m \) is the vehicle mass. The state transition equation has the following form:

\[ X_k^- = A_k X_{k-1} + B_k u_k \]
\[ P_k^- = A_k P_{k-1} A_k^T + W_k Q_{k-1} W_k^T \]

The state transition, control, and process noise matrices are respectively:

\[ A_k = \]
compatibility with the state estimate via the following:

Outliers: The incoming measurements are tested for

Sure: A strategy is needed for the rejection of

DVL: Because the DVL is prone to generating spurious mea-

ties: The navigator receives measurement inputs from three

sources: The map-based localizer for absolute position

measurements given by the MRU. These equations do not incorporate the
roll angle because it is assumed that the vehicle is stable
along that axis.

The navigator receives measurement inputs from three

sensors: the map-based localizer for absolute position

measurements, the DVL for vehicle-frame veloc-

ties, and the pressure sensor for the position z. Whenever the navigator receives new sensor inputs, the following EKF update sequence is executed:

\[
\begin{bmatrix}
1 & 0 & 0 & \Delta t \cos \psi \cos \phi & -\Delta t \sin \psi \sin \phi & -\Delta t \sin \phi \\
0 & 1 & 0 & \Delta t \sin \psi & \Delta t \cos \psi & 0 \\
0 & 0 & 1 & \Delta t \cos \psi \sin \phi & -\Delta t \sin \phi & \Delta t \cos \phi \\
0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 1
\end{bmatrix}
\]

\[B_k = \begin{bmatrix}
\Delta t^2 \cos \psi \cos \phi & -\Delta t^2 \sin \psi \sin \phi & -\Delta t^2 \sin \phi \\
\Delta t^2 \sin \psi & \Delta t^2 \cos \psi & 0 \\
\Delta t^2 \cos \psi \sin \phi & -\Delta t^2 \sin \phi & \Delta t^2 \cos \phi \\
0 & 0 & 0 \\
\Delta t & 0 & 0 \\
0 & \Delta t & 0
\end{bmatrix}
\]

\[W_k = B_k \]

Where \( \phi \) and \( \psi \) are the pitch and heading obtained from the MRU. These equations do not incorporate the roll angle because it is assumed that the vehicle is stable along that axis.

The navigator also maintains an estimate of the vehicle’s altitude based on measurements given by the DVL. Like the velocity measurements, the altitude measurements are subject to outliers. However, EKF filtering is not appropriate as we do not have an \( a \) priori bathymetry map for performing prediction. Instead, a median filter is used, with a window size of 5 measurements.

**B. Sonar-based localization**

The forward looking Tritech Micron sonar provides sensing capabilities about the outer world. It mechanically scans a variable sector (up to 360 degrees) around...
itself using sonar technology. Sparus\textsuperscript{AUV} uses the sonar to scan the surroundings and generate an occupancy grid map that we use for localization and mapping tasks. Since the sonar is mechanical, it takes several seconds to cover a full sector (depending on the resolution of the scan angle). The sonar data represent range distances from the origin of the sensor with a known angle. The sonar localization can be divided into two steps:

- Map generation
- Map based localization

1) Map Generation: To generate a map we need to reference the polar range coordinates returned by the sonar in a Cartesian world. To do that we need a robot position that it is given by the navigator component. At the beginning we only have a relative position (with respect to the start of the mission). So, each range information that we use to generate a map depend on the current position of the robot, this is because we cannot have a full sector coverage in real time, thus the movement of the robot affects the sensing. To overcome this problem, we do a first full scan exploration at the beginning of each mission without moving the robot. This permits to avoid using the position of the robot in the Cartesian world (which is unknown) and also will avoid the introduction of noise caused by the navigation sensors. After a full scan is complete we generate a map that we will use as a reference Cartesian world to calculate the robot’s position. The map generation returns a polygon that represents a rough estimation of the surroundings and from which we extract the bottom-left most corner to use as a origin of the Cartesian map.

To extract the polygon we use an algorithm that we developed adhoc for the competition. It is based on the common Hough Transform. The latter is a method for extracting lines (also other features depending on its tuning) from a set of points. In our case, the points are returned by the sonar. However, the sonar data is highly noisy, especially in certain conditions, so, we did some adaptation to generate a clear polygon out of the returned data.

At first, a general Hough Transform is performed, properly tuned to filter useless results. The result is a set of lines that represents possible features (lines) in the world. However, the Hough Transform result is not always accurate especially with highly noisy sonar data. Often multiple lines representing the same feature in the world are returned. We developed a simple algorithm to merge all the lines representing the same feature. This algorithm checks the similarity of each line with respect to the world. We define two lines similar if:

a) Their orientation difference does not exceed a user defined threshold
b) Their shortest distances between their vertexes does not exceed a user defined threshold

If two lines are found to be similar we merge them in a single line.

After applying the afore mentioned method, we extract the minimum polygon including the position of the robot (which is \([0, 0]\)) since we did not move and we are at the beginning of the mission. To extract the minimum polygon we use a method based on ray tracing.

Once a polygon is extracted, the forming lines are drawn into an occupancy grid map, big enough to contain the polygon. An occupancy grid map is a matrix where an occupied cell is 1 and an empty cell is 0. Cells with values 1 represent obstacles. This is enough to define a map estimation of the surroundings.

2) Map-based localization: The first step is used to define a map that we are going to use as base to do localization. The map is at first used to reference a cartesian world using the left-bottom most corner of the map. The localization step tries to correct the pose returned from the navigator using the sensor data. Once we sensed a map, we use the same sensor as we move to find possible poses inside a map. The idea is to use the last range returned by the sonar to generate a set of possible poses inside a map. If we receive a range from the sonar pointing at a given angle, we can generate a set of possible poses that is composed of those poses that are consistent with the last beam shot. This algorithm is also based on the Hough Space concept which is a matrix generated by the Hough Transform. A Hough Space is a voting space that tells us possible lines passing through a point. In this case each beam will increase the corresponding voting space matrix element by 1 if the returned range might have been shot from a given position. The explanation is quite tricky and it is strictly based on the algorithm by [2].

The output of the algorithm (an example of the algorithm’s outcome is shown in Fig. 8) is the most probable pose, i.e. the cell with more votes in the Hough Voting space generated after a full scan is acquired. Note that in this case we do not need to stand still in order to apply our method because we can reference each beam with respect to a fixed cartesian world. However, accuracy problems might arise. The final output is a pose hypothesis that is passed to the navigator with a relative uncertainty based on the map resolution and the sonar accuracy.
C. Gate crossing

In order to cross the validation gates the Sparus\textsuperscript{AUV} navigation system is used. That is, the vehicle navigates to an a priori known point of the competition arena: faces the gate by achieving the also known orientation and goes on to cross it. Optionally, the underwater structure detection module can be used to stop the vehicle above the structure: and then navigate to the gate based on its relative position to the structure.

D. Underwater structure and target detection

Although the mission takes place in a limited visibility environment our approach to detect the underwater irregular structure and the mid-water target relies mainly on vision techniques: though in the case of the mid-water target sonar data is used to determine its approximate position. Since the absolute positioning method explained in section V-B requires the Micron sonar to scan in the horizontal plane, it can not be used to detect the underwater irregular structure. Hence, rather than using an extra sonar to detect the underwater irregular structure, we decided to face the problem using two cameras and computer vision algorithms. A down-looking camera is used to detect the underwater irregular structure and a forward-looking one is used to detect the mid-water target, once its approximate position is determined by sonar.

Two main factors determined this decision. On one hand, the cost of using two cameras is obviously smaller than using sonars. On the other hand, Micron’s sonar readings have not enough resolution to perform precise detection of such a small targets. Hereby, we decided to put our efforts in implementing a robust vision-based strategy to overcome the difficulties of the environment. Because of the limited visibility we can not expect to identify and detect the targets from far distances. Therefore, several behaviors performing search patterns have been implemented to bring the vehicle close to them until they can be seen by the on board cameras. Two processing units have been implemented to perform the underwater irregular structure and the ball detection respectively. Both detections rely on a color segmentation algorithm. The colour segmentation is based on a mixture of Gaussians algorithm [4] which estimate probability density functions in color space, offering robust models which are able to cope with varying lighting conditions.

Once the image is segmented, the two processing units operate in a different way.

1) Underwater structure detection: The underwater irregular structure detection algorithm is based on determining the structure’s overall shape by accumulating evidence over a large number of images. This is in contrast to last year’s algorithm which focused on estimating and tracking the pipe’s local orientation on a per-image basis. We believe that this approach is better suited to handling varying changes in the pipe direction.

From the segmentation results, a blob labeling algorithm is applied, and the largest blob is selected, as long as it satisfies some simple shape and minimum area requirements. Using the camera’s projection matrix and the navigator state, a globally-referenced evidence grid is projected onto image plane. The cells which lie inside the image are then incremented or decremented depending on the degree to which they contain pixels from the selected blob. By limiting the per-image operations to these simple morphological tasks and foregoing costly algorithms such as the Hough Transform, images can be captured at a higher frame rate and thus the robot can traverse the mission area more rapidly, taking advantage of its good hydrodynamic performance.

A further difference from last year’s strategy is that the underwater irregular structure following task will be postponed to near the end of the mission. The vehicle will continuously capture images as it performs other tasks to populate the evidence grid, keeping track of areas that have been mapped. When the time arrives to perform the underwater irregular structure following task, one final survey trajectory will be dynamically generated to complete missing areas of the evidence grid in the region of interest.

With the grid complete, it is processed with thresholding and morphological operations. Then the largest blob
is found as the pipe. A trajectory can be generated to follow the pipe by applying thinning operations to this blob.

When the mid-water target is sufficiently close (i.e. it is visible in the camera’s field of view), a blob labeling is applied on the segmented image and the blob having the highest circularity is selected. This circularity measurement is computed using the axis-aligned bounding box of the blob and its expected circular mass. After the first successful detection, the operation is repeated within a tracking window around the object to speed up the computation in the successive frames.

2) Mid-water target detection: The mid-water target detection is faced in two steps: first, its approximate location is obtained by means of sonar data processing; then, computer vision techniques are used when the target is in the camera’s field of view.

The generation of a map by means of sonar data also allows to perform accurate buoy detection (i.e. mid-water target detection). The latter, tries to find possible targets using again the sonar data returned by the Tritech Micron sensor. To do that, the information beam returned by the sonar are passed to a Sonar Target Detector component that listens and record range beam information. The first task is to acquire a map. This is done requesting the map to the sonar-based localization system and waiting until it is available. Once the component has a map, it starts listening to sonar data and referencing the points inside the map it just acquired. The reference is done using navigation data provided by the navigator which is absolute with respect to the outer world defined by the occupancy grid. The use of a map is important to easily discard all those points returned by the sonar that are sound reflection and only cause noise in the range image. After a full scan and proper filtering we have a clear set of possible targets laying inside our occupancy grid map. We can clearly discard all those points that lay nearby the walls. At this point a contour detection based on blob search is performed. This method returns a set of possible targets in \((x, y)\) coordinates referenced to the occupancy grid.

2) Target releasing: Using the centroid coordinates obtained from the target detection, our target releasing system uses a PID controller in the surge and yaw DOF to keep the target’s centroid centered on the image. Once it is centered, the target is approached using a constant setpoint in surge until its area goes beyond a certain threshold. After that, the depth positioning behavior is used to move the vehicle one meter down. Finally, the cutting system is triggered on and the vehicle moves forward in order to cut the rope.

F. Wall surveying

In order to complete the wall following task, it is required to sense the distance of the vehicle with respect to its surroundings. Among the sensor suite of Sparus\textsuperscript{AUV}, two sensors are able to provide that information: the imaging sonar and the echosounder. For surveying the wall, the Micron sonar is set up to scan just a sector of 90 degrees on the right side of the vehicle (see Fig. 11). Using this configuration each sector scan is obtained at a frequency of approximately 0.4Hz, which allows us to
receive updates of the environment quickly enough to control the vehicle for this task. The second sensor, the echosounder, is placed at the front part of the vehicle and returns the forward distance at a frequency of 1Hz.

The procedure to perform the wall following task using these two sensors and taking into account that the vehicle is only actuated into surge, heave and yaw is detailed here (fig.11):

1) Approach the wall frontally until the distance given by the echosounder is smaller than the wall following distance (i.e. 3 meters).
2) Turn the vehicle left 90 degrees.
3) Using the data gathered by the imaging sonar, find the maximum value in each beam and using its angle and distance compute a 2D point with respect to the robot.
4) From the set of points obtained in step 3, fit a line using least-squares.
5) Using the slope of the obtained line as well as the distance to it, correct the vehicle yaw and move forward.
6) If the echosounder reads a value smaller than the wall following distance the procedure returns to step 2.

Fig. 11: Schema of the wall surveying approach using the echosounder and the Micron sonar beams (in blue). Red crosses depict the maximum return points detected by the sonar and the corresponding fitted line appears in green. Its slope is used to correct the heading of the vehicle.

G. Pinger localization

We propose to employ a two-hydrophone array to localize the target equipped with an acoustic pinger. A priori knowledge of the depth of the acoustic pinger allows to disambiguate the Direction of Arrival (DOA) of the acoustic wave when employing less than three hydrophones in a three-dimensional environment. To generate an estimate of the location of the beacon, a particle filter is employed.

An example of the possible locations of the pinger given a single noise-less Time Difference Of Arrival (TDOA) from the hydrophone array is shown in Fig. 12.

Fig. 12: Hyperboloid (blue), z-plane cut (green) and 2D hyperbola (red).

Moreover, Fig. 13 shows different regions of possible pinger locations when TDOA quantization is considered. It can be observed that multiple measurements at different locations must be fused in order to generate a bounded estimation of the location of the pinger.

1) Audio processing pipeline: A block diagram of the audio processing pipeline is shown in Fig. 14. The pre-amplifier conditions the signal for the Analog to Digital Converter (ADC), which samples the audio signal from both hydrophones simultaneously. Once digitized,
Fig. 14: Hydrophone audio processing pipeline

the software is in charge of applying the beamforming method [5] to the acquired data to determine an estimate of the time delay between the two signals.

Regarding hydrophone placement, it must be taken into account that, due to spatial aliasing, the maximum baseline $b$ is dictated by

$$b_{\text{max}} = \frac{\lambda}{2} = \frac{c}{2f}$$

where $c$ is the speed of sound and $f$ is the frequency of the acoustic wave and in this case the frequency of the monochromatic tone emitted by the pinger. In the case of a sinusoid of 12 kHz, we propose to employ a baseline of 6 cm.

The presence of the tone burst is detected by means of a matched filter. Moreover, prior knowledge of the duty cycle of the sinusoidal burst is employed to detect possible outliers.

2) Particle filter: The time delay estimates generated by the pipeline presented in Fig. 14 are fed into a particle filter to render an estimate of the location of the catamaran. Due to the discretization of the measurements with the chosen sampling frequency, assuming measurement noise as Gaussian is not appropriate, preventing the use of Gaussian parametric filters such as the Extended Kalman Filter.

Pinger position and velocity are parametrized as

$$\mathbf{x} = \begin{pmatrix} x & y & v_x & v_y \end{pmatrix}^T$$

where $\mathbf{r} = (x, y)$ is the position at the expected depth and $(v_x, v_y)$ is the corresponding velocity in the same reference system.

In order to deal with outliers, the technique introduced by Maiz et al. [6] is employed. This allows to reject incorrect measures that are likely to arise due to factors such as acoustic interference and reduced coherence [5].

As the Sparus$^{\text{AUV}}$ navigates the environment, pinger emissions are processed and incorporated in the particle filter, generating an estimate of the location of the pinger. At the point where the catamaran tracking task begins, the AUV performs a slow yaw rotation in order to refine the first pinger location estimate. Once this is done, Sparus$^{\text{AUV}}$ starts the tracking phase that allows to follow the catamaran, as explained in the SAUC-E Mission Rules.

VI. INNOVATIONS

Sparus$^{\text{AUV}}$ has been built from scratch for SAUC-E’10 contest, and the vehicle as a whole represented a big innovation in the Underwater Robotics Laboratory of Girona, providing a new testing platform, which offered new experimentation opportunities, and that showed to be a good platform in a number of experiments held during this year. In this section we want to highlight some of the introduced advances on Sparus$^{\text{AUV}}$, and especially to underline the ones made for SAUC-E’11.

A. Innovations in mechanical design

Sparus$^{\text{AUV}}$ mechanical design manages to incorporate all the necessary sensor suit in a compact size and low weight vehicle. The number of thrusters as well as the sensors have been specially selected to optimize the weight while still offering a vehicle with complete capabilities to be a functional research platform. All the sensors are easy to reach and the main electronics can be accessed by removing only the tail cone. The separate battery housing minimize downtime between missions by allowing a quick interchange with other fully charged battery pack, without the need of opening the pressure housing. Also, it assure safe handling and charging of the NiMH batteries. The WiFi antenna can be easily extend by 5 meters without opening the skin allowing to communicate with the vehicle while submerged.

B. Innovations in electronics design

The main innovation here regarding our previously developed vehicles is the selection of an state of the art computer combining high processing power with low consumption. Also, the integration of heat pipes help to quickly transfer the heat to the aluminum housing and maintain the appropriate temperature. In the electronics part, constant monitoring of the internal state of the pressure housings assures early reaction in case of a problem.

C. Innovations in software architecture implementation

Compared to last year, the utilization of the ROS platform to implement the control architecture brings us the opportunity to use visualization tools and other software components that satisfy some of our needs and that are available off-the-shelf. It also allows for
trivial communication with other ROS packages and good documentation and support.

**D. Innovations in pinger detection**

The advanced method used for detecting the range and bearing of a pinger using hydrophones implies the innovative ability to face tasks relying on a sound source with accuracy.

**E. Innovations in pitch control using fins**

The design and development of a fins system to control SparusAUV pitch DOF brings about a new way for the vehicle to navigate, making it able to reach waypoints in a more flexible way.

**VII. FINANCIAL SUMMARY**

Figure 15 summarizes the list of expenditures involved in the participation of the VICOROB-UdG team to SAUC-E’11. The expenditures show the cost of the entire robot, built for last year’s competition, considering all materials and devices that can be found in the vehicle. Also, the estimated cost of the mobility of 7 students and 3 supervisors to La Spezia (Italy) from July 4th till July 10th, have been included. Regarding the incomings, most part of the expenses have been financed by research projects of the VICOROB research group. Also different sections of the University of Girona have collaborated with the team, as well as local institutions and companies from Girona nearby. We thank all of them.

**VIII. RISK ASSESSMENT**

SparusAUV construction was motivated by the SAUC-E competition (low depth, restricted working area) but also to serve as a future research vehicle (deeper and longer missions). Construction, operation and maintenance of such an AUV requires the handling of sharp and power tools, raw materials, chemicals, energy and gases under pressure and possible robot physical reactions without warning or under no direct human supervision. It is essential that all of the potential hazards have to be identified and minimized, standard safety procedures to be followed and non-standard vehicle-specific safety procedures to be produced. In our case, two major risk lines have been identified for the life time of the SparusAUV: a) construction and maintenance, b) operations.

**Fig. 15:** Summary of expenditures for SAUC-E’11 competition, including the 2010’s robot construction.

**A. Construction and maintenance**

The construction of the 50 meters depth rating pressure housings required special mechanization. For this reason, we did the calculations and the schematics but they were constructed in specialized workshop. Whenever chemicals, heat, sharp or power tools were used, necessary protection clothing, ventilation and standard safety procedures were applied.

**B. Operations**

Potential hazards in the operations can affect the AUV and/or humans in direct or indirect contact.

1) **Risk for the AUV:**

- Because of the SAUC-E competition nature we assume that the vehicle cannot be lost, either in surface or at the bottom and no special precautions have been implemented. The competition area is constrained at three sides by the harbor walls and at the open sea side by a safety net. The area is
shallow and the vehicle is well supervised by means of acoustic transponder, surface vehicles and divers.

- In case of power loss, the vehicle is positive buoyant and eventually it will surface.
- All pressure housings are constantly monitored internally by pressure, humidity and water leak sensors. In case of water leak or extreme humidity the AUV will abort the mission and surface.
- All electrical parts are well insulated and fuses are protecting the main batteries from short circuits.
- Voltage and current are monitoring constantly. In case of abnormal consumption or low voltage the mission is aborted and the vehicle surfaces.
- There are not any penetrating cables in the housings. All the cables are ended with connectors, so in case of cable failure the water leak will stop at the connector end and will not enter the housing.
- Aluminum skeleton, plastic skin with buoyancy foam and metal reinforcements are protecting the sensors from potential collisions.

2) Risk for the human:

- Although the weight of the AUV is less than 35 Kg, moving base and crane are utilized all the times for transportation, deployment and recovery. That way, injuries from accidental droppings and back problems are minimized.
- Propellers are protected with solid plastic net that does not allow the contact with fingers. Precautions must be taken for avoiding long hair and loose fabrics in the proximity of the thrusters.
- The cutter is designed to expose a very small area, not enough for a finger to enter. Moreover, whilst outside of the water and when not needed, a protective cap is placed.
- In case of abnormal behavior, an easy to access external switch can cut the power from the batteries.
- No harmful voltage. The vehicle is powered by two DC circuits of 12 and 24 volts with NiMH batteries packs.
- It is well known that improper charging/discharging cycles of NiMH batteries can produce flammable gases that under pressure can potentially be explosives and special precautions has to be taken if used in hermetrical sealed housings. We address this problem by:
  - Including a venting screw in the housing’s cap.
  - Using state of the art chargers.
  - Monitoring the housing’s internal pressure.
  - Separate housings for batteries and electronics.
- Non sparking components are used (only solid state relays, etc).
- Dedicated procedure is designed for charging the batteries and opening the housings.

IX. CONCLUSION

This paper has presented SparusAUV as an entry to SAUC-E’11, a vehicle designed and developed by the VICOROB-UdG Team to face the SAUC-E challenge. Since January 2011, the VICOROB-UdG team of students has intensively worked to develop a totally operative and reliable software architecture and solutions for the proposed mission tasks. The main principles of design have been reported together with the redesigned software architecture and the proposed solutions for every mission task.

REFERENCES

Enric Galceran obtained the M.Sc degree in Computer Science in 2010 from the University of Girona (UdG), Spain. He is been working as a software developer at the Underwater Robotics Lab at the UdG as a scholarship holder since 2008. From 2009 to 2010 he also worked in Constraint Satisfaction Problems at the Software Engineering, Logic and Programming Lab at UdG. In summer 2010 he took part in SAUC-E as the software manager of the VICOROB-UdG Team. He is now a PhD candidate at UdG and his research interests are focused on coverage path planning for underwater vehicles. He also likes travelling, appreciating a good work of art, cooking and practicing any kind of sport.

Carlos Becker holds a degree in electronic engineering from the University of Rosario in Argentina and has recently graduated from the European Master in Computer Vision And Robotics (ViBot). His master thesis was directed towards single-beacon acoustic localization and navigation, conducted at the underwater robotics lab of the University of Girona. He is interested in many fields such as underwater acoustic localization, navigation and image processing. Besides being passionate about his field of research, he also enjoys music, playing the guitar and doing activities with friends in his free time.

Chee Sing Lee hails from the great state of Oregon where he discovered his passion for robotics participating in FIRST Robotics competitions during high school. He went on to become a mentor for the program and completed his B.Sc. with honors in Electrical and Electronics Engineering at Oregon State University. Currently, he is wrapping up an Erasmus Mundus Masters in Computer Vision and Robotics (VIBOT) between Heriot-Watt University, Universitat de Girona, and the Université de Bourgogne. He will be continuing at UdG as a PhD student pursuing his love of SLAM and autonomous navigation as a part of the VICOROB group. In addition to saving the world with robotics he also enjoys travelling, board games, and a good joke.

Simone Zandara comes from the University of Bologna Italy where he finished a bachelor and master degrees in Computer Science. During its master degree he focused on artificial intelligence and finally discovered its interest for robotics during a funded abroad research instance in Australia. During this period, he studied SLAM and scan matching which were later subject for his master thesis. After graduation he looked to stay in the robotics field and was hired by the University of Girona as a PhD student to purse a thesis in Autonomous Underwater Navigation. His main work is sonar based sensing to use with SLAM methods. He is also a keen traveller with passion for movies and music.
Arnau Carrera obtained his B.S.c in Computer Science back in 2009 from the University of Girona and is now pursuing the M.S.c. degree in computer science in 2011 from the University of Girona (UdG), Spain. Currently, he has a scholarship to collaborate with the Research Center in Underwater Robotics (CIRS) of Girona since 2010 as software developer. His final project in M.S.c is based on adapting the previous AUV's software architecture to the ROS (Robot Operating System) framework.

Pere Pares obtained his B.S.c degree in computer science back in 2009 from the University of Girona and is now pursuing the M.Sc of the same speciality. He joined the Computer Vision and Robotics research group (VICOROB) of UdG in 2010 and he's been working in the Research Center in Underwater Robotics (CIRS) of Girona as an AUV's software architecture developer and maintainer until today. His research interests mainly comprise software engineering and anything related to the field of robotics or computer vision.

Carles Candela is finishing the studies of Electronic Engineering in the University of Girona. He has joined the Research Center in Underwater Robotics (CIRS) of Girona this year to make his BSc degree final project. His project is about the design and construction of fins for controlling the pitch of the Sparus vehicle and the design of the control system. He has worked for three years as an industrial robot installer and programmer.

Marc Carreras received the M.Sc. in Industrial Engineering (1998) by the University of Girona, Spain. Ph.D. in Computer Engineering (2003) by the University of Girona, Spain. His research activity is mainly focused on robot learning and intelligent control architectures of autonomous underwater vehicles. He joined the Institute of Informatics and Applications, University of Girona in September 1998. Currently, he is an associate professor with the Department of Computer Engineering of the University of Girona and member of the Research Center in Underwater Robotics (CIRS) of Girona. He is involved in National and European research projects and networks about underwater robotics.