SAUC-E 2012 - The HANSE Team

Jan Hartmann, Thomas Tosik, Jannis Harder, Jan Fechner, Hans-Joachim Reddecker, Andrea Kampsen, Patrick Zenker, Sven Friedrichs, Cedric Isokeit, Patrik Stahl, Peter Hegen and Erik Maehle

***Abstract*—This paper describes the University of Lübeck's entry to the 2012 SAUC-E competition. As in the previous three years, our AUV HANSE has been improved by students in practical courses with a strong focus on the SAUC-E competition. HANSE is mainly built of commercially available commodity products. Professional thrusters, sonars, and AHRS system have been added in the previous years. Two hydrophones and an industry camera have been further added for SAUC-E 2012.**

As we are joining the SAUC-E for the fourth time since 2009 with few hardware alterations, this year's focus is on software innovation. Our own framework was replaced in favour of the popular Robot Operating System (ROS). Further improvements have been developed in the fields of localization, behaviours, pinger detection and simulation environment, which will all be presented in this paper.

I. INTRODUCTION

The main goal of the HANSE project is to encourage our student's interest in robotics in general and the challenges of underwater robotics specifically - and of course to build a robot which provides the features of robustness and expandability necessary for a project designed to run for several years. Our notion is that this can be best achieved in a competitive environment.

The AUV HANSE (see Fig. 1) has thus been developed specifically for the SAUC-E in a series of practical courses (bachelor as well as master) and several master thesis' by students over the last four years. HANSE's main housing, a waterproof Peli case, is mounted to a Polypropylene (PP) base frame. A number of Buccaneer connectors provide a generic interface from the controlling laptop to thrusters and sensors, which are also mounted to the base frame. This ensures the expandability of the design concerning sensors as well as control hardware.

The robot control software runs on a 12.1" standard laptop. A modular Qt-based[1]software framework was developed till 2011 but has been replaced with the popular robotics framework ROS (Robot Operating System)[2] in 2012. For more details see Section II-C.

The HANSE robot won the innovation prize in the SAUC-E 2009, where the focus was on building the robot itself and the handmade thrusters. In 2010, the latter were replaced by the more robust SeaBotix thrusters and a scanning sonar was added. Further, the custom software framework was developed and, due to changes in the mission rules, new algorithms were developed.

For SAUC-E 2011, the framework was further improved, navigation algorithms and an Attitude Heading Reference System (AHRS) were added. A simulation environment was developed, which allowed for a faster development process. In the SAUC-E 2011 HANSE achieved the first prize.

For the fourth HANSE iteration a switch to the robotic framework ROS was decided, in hope to benefit from open source aspect and re-usability of software developed for ROS. A large part of this year's work was therefore directed into migrating existing software.

In terms of algorithms, a vital part of the system, the localization, could be further improved. A new pinger detection algorithm was added. Further, new features have been added to the simulation environment.

The remainder of this paper will be structured as follows. A general description of the robot will be given in Sec. II. First, the robot hardware will be presented in more detail in Sec. II-A. In Sec. II-C the new framework switch and the behaviour creating will be discussed. In Sec. III innovations to the software architecture, the main algorithms and behaviours will be presented. A financial summary and risk assessment will be given in Secs. IV and V. Then hard- and software will be evaluated in comparison to the last three years to give conclusion on this year's project innovation. Finally, the team members and their responsibilities within the team will be introduced.

II. DESCRIPTION

A. *Overview and mechanical actuators*

The base frame of our AUVs owes the form to a sledge. This form was chosen because the thrusters can be attached at any position on the sides of the scaffolding. Thus the position of each thruster can be evaluated during the test runs, and can be mounted to its optimal fixing point. Another advantage of this form is that the AUV can be carried conveniently by two people. The base frame of the AUV is made out of 50 mm Polypropylene (PP) tubes. We have chosen this material because of its light weight and the possibility of welding single parts together easily. To increase the solidity of the frame it was strengthened by glass fibre sheathing. Additional holes, which are drilled in the frame in distances of 10 cm apart, allow the flooding of the frame, so it has near neutral buoyancy.

On the base frame a waterproof case (Pelicase 1400) is fixed, which is the main pressure hull of the robot.

[1]http://qt.nokia.com
[2]ROS, http://www.ros.org

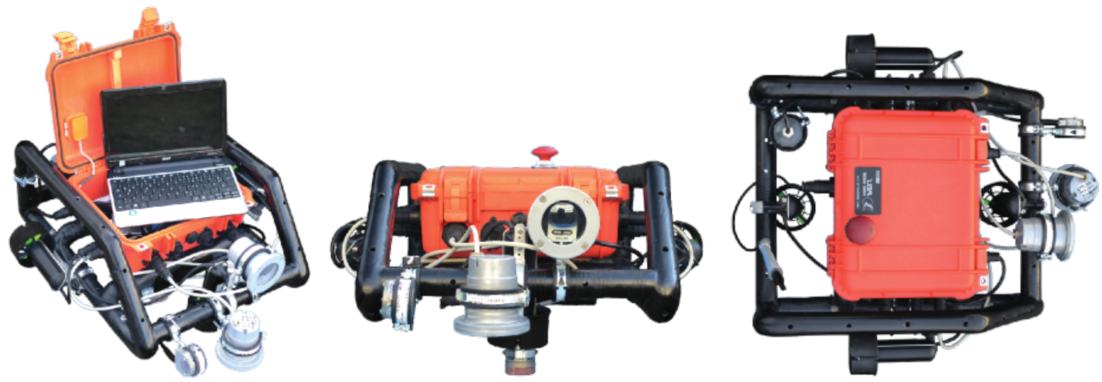


Fig. 1: The autonomous underwater robot HANSE.

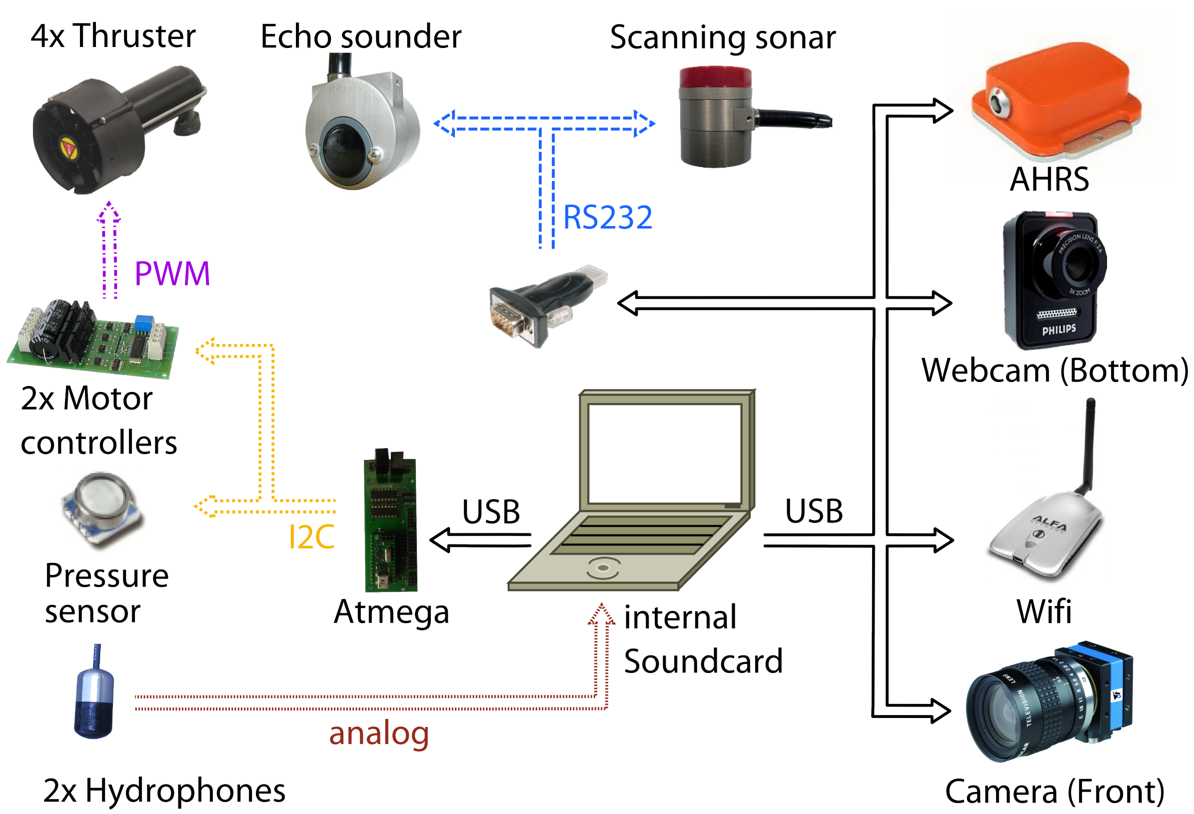


Fig. 2: Hardware overview of the HANSE AUV.

An internal frame, made of steel and wood, makes the relatively soft case inherently stable against expected amounts of pressure. With its inside measurements of 30x22.5x13.2cm the case offers enough space for most electronic parts and power supply.

Several modules like thrusters and sensors are placed outside the main case. Their exact mixture and position on the frame can be adapted to the actual mission. All modules are attached by waterproofed BULGIN Buccaneer (PX0748/P) connectors to the main case. To ensure stable horizontal orientation of the robot, external weights can be attached on to the frame. The HANSE AUV in the current version can be seen in Fig. 1.

1) Motors: For propulsion four SeaBotix BTD150 thrusters are attached on the main frame. Two are placed on the sides for horizontal movement and rotation around the yaw axis. The other pair is placed in front and back of the main body and allows diving. Because of positive buoyancy the thrusters must work to submerge.

2) Camera housing: As camera housing for our cameras we use a lamp housing that is usually used for illuminating garden ponds. These cases have a 5 mm glass panel, and are waterproof up to 10 m.

B. Electronic design

The general electronic design can be seen in Fig. 2. Each component will be introduced in detail in the following sections.

1) Power Supply: The HANSE power supply consists of three electrical circuits. The first contains the notebook, the central processor unit, and one webcam connected over USB. The second circuit is providing power for the sonars. It contains two small serial connected lithium-polymer-accumulators with 1000mAh and 11.1V each.

The third, the main power circuit supplies all remaining modules. It is powered by a three-cell high performance lithium-polymer-accumulator providing 10 Ah at 11.1 V. The battery is secured by a fuse before its first connector. This circuit can be disconnected by the kill switch, located on the top of the case. Pressing it interrupts the electricity supply of the engines immediately, and the AUV emerges.

2) Main Computer: As the central processing unit we use the ACER "AspireTimeline1810TSpecial Edition" notebook. It contains an Intel(R)Core 2 Duo processor SU7300 (1, 2 GHz), and 500 GB hard disc. With a width of 285 mm we have around 5mm space between the notebook and the case at the left and right side. This demands careful space management inside the case.

3) Xsens MTi/AHRS Sensor: As and IMU-Unit we use the Xsens MTi/AHRS Sensor. The Xsens MTi is connected to the system by USB, which allows us to read the sensor at 1 Mbps. It provides the system with reliable attitude and heading informations, that is used inside the navigation and localization algorithms and to correct sonar images.

4) Pressure Sensor: To measure the actual depth, we use a 'MS5541-CM' pressure sensors from Intersema. It has an absolute pressure range from 0 to 14 bar put out as 16 Bit value and achieves an accuracy of 2 cm. It is connected over a self built SPI-to-I2C translator that is located in the housing of the pressure sensor. This translator additionally computes the temperature compensated pressure data as described in the datasheet of the sensor and gives the result to the I2C-Master.

5) Sonar Modules: We use the "Model 852 ultra-miniature scanning sonar" from Imagenex for localization. This sonar has a beam width of 2.5 degrees x 22 degrees. Adjusting the gain, ranges from 150 mm up to 50 m are reachable. Additional it has two step sizes: normal (3 degrees) and fast (6 degrees). With a maximum range of 50 m one rotation requires 16 seconds in case of the normal mode and eight seconds in case of the fast mode. The sonar can work with 675 or 850 kHz. We are using 850 kHz to minimize the cross noise with the second sonar.

The second sonar module, the "Model 852 ultra-miniature echo sounder", uses 675kHz as working frequency. It has a conical beam of 10 degrees width, and range scales up to 50 m. We use it primary for wall detection, so it is oriented to the port side.

Both sonar modules are connected over an RS232 serial interface. To avoid noise from the Motors we shield the sonar modules by using an "Expert Opto-Bridge" optical coupler module (Gude) for communication and a separated power supply.

6) Cameras: We use one sponsored DFK 22AUC03 camera from "The Imgaging Source" and one USB webcam 'SPC1030' from Philips. The first is facing forward primarily for ball detection. The second camera is mounted facing downwards for pipe detection. The "SPC1030" webcam grabs 640x480 pixel images with a frame rate of 5 Hz. It has a lens view angle of 80 degrees that is decreased by the water to 60 degrees. The DFK 22AUC03 camera features image sizes up to 744x480 pixel and frame rates up to 150 depending on the chosen resolution.

7) Pinger detection: In order to detect the 13kHz pinger two Aquarian H2a hydrophones are mounted onto the 'sledge'. They are then processed by the internal notebook soundcard. The orientation computing is done by the laptop, that estimates the direction by analysing the different times of arrival (TOA) of the two hydrophones.

8) Bus Network & Universal Interface Device: The communication interface between external modules and the notebook is done by our self-built "Universal Interface Device" (UID). As hardware platform we use a small ATmega168-Board from chip45, but the UID architecture is not determined to this board, it can be used for almost any type of Atmel 8-bit processors. The UID is connected by USB and is addressed by a serial interface with a configurable speed from 2400 bps up to 2 Mbps. The standard communication speed is set to 115200 baud in order to allow the using of a normal terminal program to communicate with the UID. To buffer the incoming and outgoing serial data, a 256 Byte ring buffer both for receive and transmit unit is implemented. Beside the I2C and SPI communication, additional features like GPIOs, 8 ADC channels, a small servo-controller for up to three servo motors as well as RS485 Transceiver are implemented.

C. Software

1) ROS Framework: For the SAUC-E 2012 we decided to change from our custom Qt-based framework to the Robot Operating System (ROS). ROS is an TCP/IP based modular and distributed communication framework for robotic applications looked after by Willow Garage. The user implements self running code units, called nodes, and communicates between them through so called topics. ROS features different visualizing and debug tools which simplifies the development process. Additionally to that there exists hundreds of user packages with drivers and algorithms common in the robotics world. Several other SAUC-E teams switched already to ROS, like Nessie VI (Heriot-Watt University), UWESub (University of the West of England) and DELPHIN2 (University of Southampton),

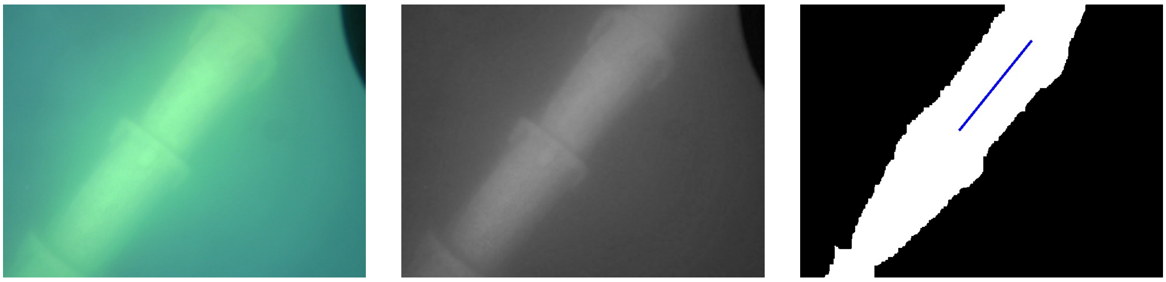


Fig. 3: Object detection using segmentation. Left: RGB image. Center: segmentation channel (red). Right: Otsu thresholding. The blue line indicates the position and direction of a yellow pipeline.

which made our decision easier. Further we hope to exchange our code easily between the HANSE AUV and the SMART-E AUV.

2) Algorithms: The main algorithms that are used on HANSE are localization, object detection, and pinger detection. Of these, the localization is most vital, as it is used in most missions and to move between missions. A lot of work has therefore been put into optimizing the particle filter-based localization algorithm that was developed for SAUC-E 2011. The main improvement lies in an increased update rate of $10Hz$ using scanning sonar, IMU, and thruster data only. The improved localization algorithm will be presented in Sec. III-C.

At SAUC-E 2011, a dedicated hardware circuit and three handmade hydrophones were used to detect the pinger. This year, a different approach, using only two microphones, was developed to simplify the pinger detection. The pinger detection algorithm will be presented in in Sec. III-D.

Having had good experiences with the color-based object detection algorithm of the past two years, the visual object detection was not changed. Each image is segmented in the appropriate color channel (e.g. red for the pipeline) using Otsu's algorithm for automatic threshold selection [1]. This method provides an efficient object detection, being fairly robust to image noise and lighting.

The presence of an object is decided based on the number of pixels belonging to the object class and the mean position of these pixels. The object orientation θ can further be found using centralized image moments [2] of the segmented image.

D. Behaviours

All of our behaviours are written with the help of SMACH[3]. SMACH stands for state machine and is a ROS independent library for building of hierarchical state machines. With SMACH its possible to build easily complex state machines. Its Phyton-based which allows us to design and generate state machines very fast. With additional tools like the *smach_viewer* its easy to debug and show the written state machines.

[3]SMACH,http://www.ros.org/wiki/smach

III. INNOVATION

A. ROS Integration

In the past years we developed an custom-made Qt-based framework. Several code units were connected through the signal and slot system from Qt. The structure resembled a direct-acyclic graph. Though the framework was working it cost an considerable amount of work time to implement several features needed in the development process. Further, we couldn't exchange parts of the software and knowledge easily between different projects.

In this year, we are using ROS as our base framework. ROS provides immediately access to wide-range of written code and debug tools. In Figure. 4, an overview of the software architecture is displayed. Each logical unit, e.g. a behaviour, was implemented as a ROS node and can easily be updated, exchanged, or stopped. Nodes communicate than via topics. On the first level *Drivers*, basic sensor, and actuator drivers, like the sonar and camera, were implemented. Here, we utilize several existing ROS nodes, e.g. an AHRS driver.

On the second level *Low-Level-Control*, the node *EngineControl* is responsible for the actuation, depth, and orientation controlling. All behaviours must use this node to move the AUV. Also the localization and some pre-processing for the behaviours are located on this level. On the third level *Behaviours*, several basic behaviours are implemented using the SMACH library. The *BehaviourControl* node starts and stops the basic behaviours.

B. Behaviours with SMACH

Behaviours in HANSE are modeled as state machines. For state machines, the ROS framework integrates the python-based library SMACH, which provides fast prototyping for hierarchical state machines. Furthermore, SMACH provides a runtime visualization of the state machine, which greatly improves the debugging process.

Generally, the overall behaviour is governed by a global state machine, which starts and stops basic behaviour state machines depending on the current mission. Two basic behaviours are shown in Figs. 5

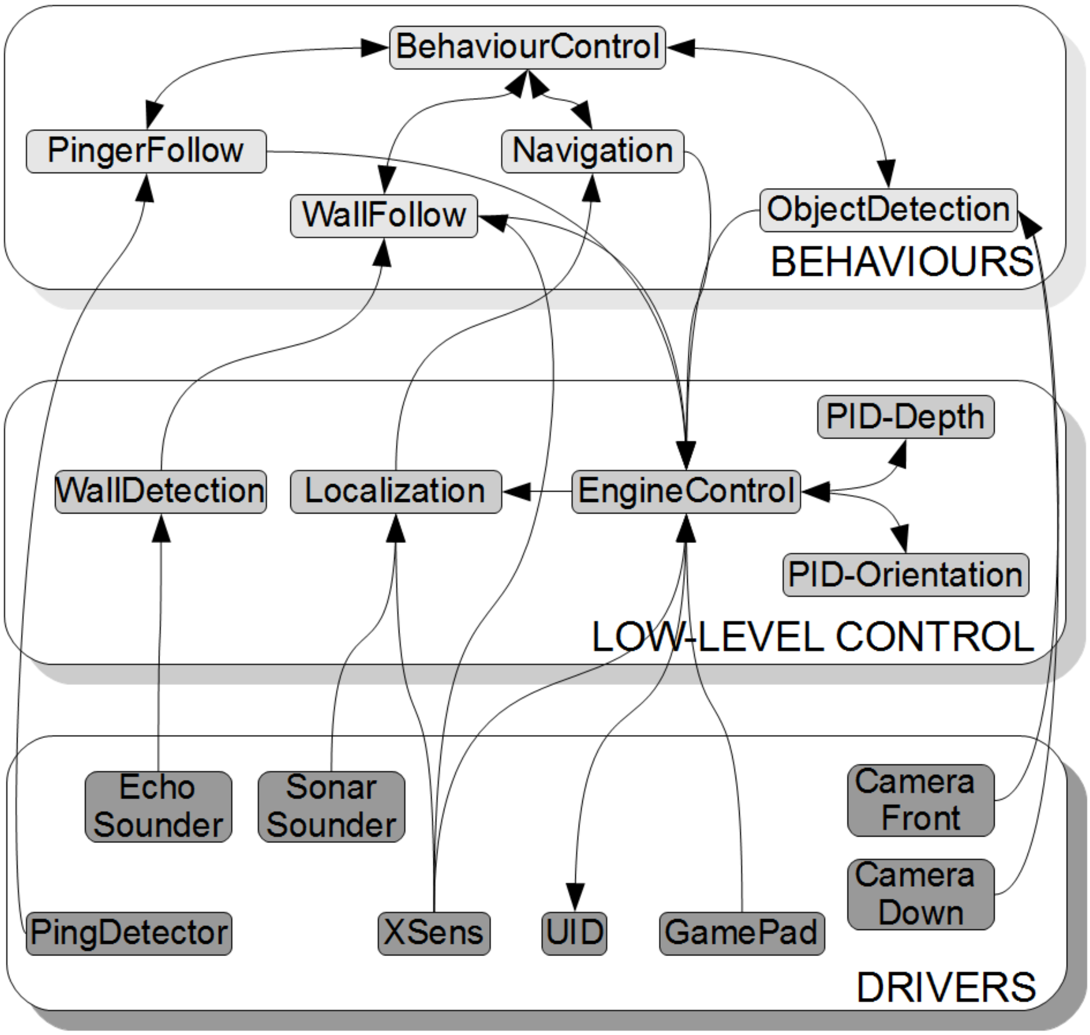


Fig. 4: The three-level software overview for HANSE. Each part represents a ROS Node. Arrows represent a communication between nodes through topics.

and 6: navigation and wall following. The navigation behaviour is, as in SAUC-E 2011, designed in a "point and shoot" manner, keeping it as simple as possible. When a new goal is set, first, the robot is moved to the target depth. Then, until the goal is reached or the goal is aborted, iteratively, the heading towards the goal is adjusted and the robot moves forward until a new robot position is estimated.

The wall following behaviour consists only of two states. In the initial state, no wall is observed by the echo sounder and the AUV rotates until a wall is deteced. In the second state, the wall is being followed. This behaviour is realized by minimizing the difference between the desired distance to the wall and the actual distance (measured by the echo sounder) using a PD controller. The controllers output is used to control the angular speed of the AUV making it move towards the wall if it is too far away and vice versa. All parameters, e.g. the desired distance and controller parameters, are configurable during run time using *dynamic_reconfigure* ROS stack[4].

C. Localization

The performance of the localization algorithm [3] was fairly robust at the SAUC-E 2011. Yet the position update rate was limited, because full 360°scans were analyzed. In 2012, using an improved particle filter implementation to allow for larger number of particles and incorporating thruster control values, the localization algorithm is now able to add single sonar readings. The position update rate is thus improved from $\frac{1}{6}Hz$ to $10Hz$ while maintaining the same accuracy. A detailed description of the localization algorithm will be given in this section.

a) Feature Detection: The localization algorithm uses a feature-based approach. Here, a feature is equivalent to a wall, as walls are easy to identify and present in most man-made environments. Such features have a characteristic signature in a sonar image: they show

[4]http://www.ros.org/wiki/dynamic_reconfigure

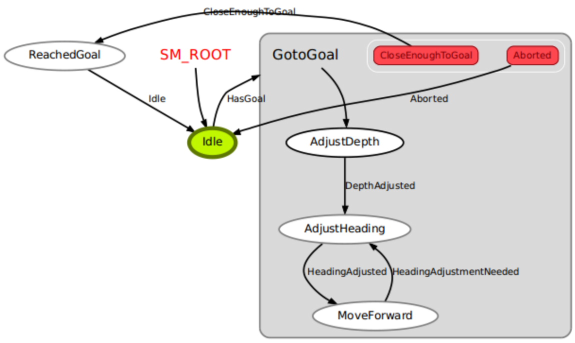


Fig. 5: SMACH state machine for the navigation behaviour.

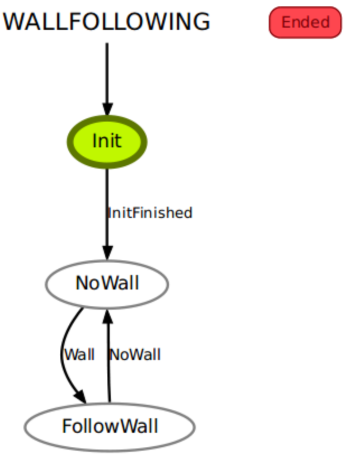


Fig. 6: SMACH state machine for the wall following behaviour.

a strong echo at and before the location of the wall and a significant drop in intensity behind the wall. The task of the sonar image feature extraction is the identification of walls in single sonar beams. It is largely similar to the gradient-based filter that was used in 2011.

To enhance the regions of the sonar image characteristic to walls, a multi-scale gradient filter is used. 1D Haar wavelet responses at different scales are multiplied for each beam pixel to form the beam gradient G as

$$G(x) = \prod_{k \in K} \left(\sum_{i=x-k}^{x} B(i) - \sum_{i=x+1}^{x+k} B(i) \right), \quad (1)$$

where x is the distance from the robot, K is the set of all scales to be evaluated, and $B(i)$ is the echo intensity at distance i. The Haar wavelet responses can be efficiently calculated using integral images.

Non-maximum suppression is now applied to the beam gradient to identify potential walls. Further heuristics, concerning gradient magnitude and neighbouring wall features based on assumptions on continuity of walls and minimum lengths of wall segments, are applied. This introduces a delay of one frame.

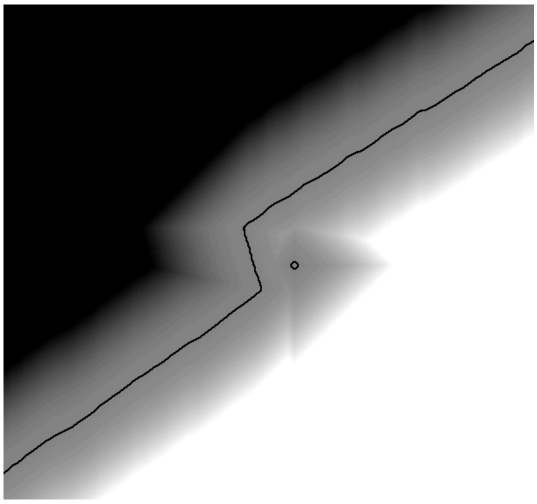


Fig. 7: The pre-calculated distance map. White and black values represent large distances (black when inside walls).

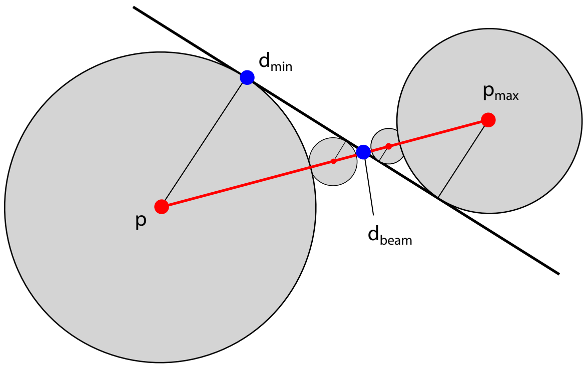


Fig. 8: Rather than considering the nearest neighbour d_{min} of a wall feature p, the nearest neighbour along the sonar beam (red line), d_{beam} is estimated. This is solved iteratively starting at point p and the point p_{max}, which is defined by the maximum sonar range along the beam. Using the distance map and a combination of bisection search and ray marching, d_{beam} can be efficiently estimated.

b) Particle Filter: A particle represents a potential position of the robot in the environment, i.e. map, in the particle state s. The state of the particle m at time t now reads:

$$s_t^{[m]} = (x, y, \theta, \nu_x, \nu_y)^T, \quad (2)$$

where (x, y) is the 2D position, θ the heading, and (ν_x, ν_y) the linear velocity.

For each feature detected by the scanning sonar, each particle state is now updated using orientation data from the IMU:

$$s_{t+1}^{[m]} = (x + \nu_x, y + \nu_y, \theta + \theta_{IMU}, \nu_x, \nu_y)^T + \mathcal{N}(0, \Sigma) \quad (3)$$

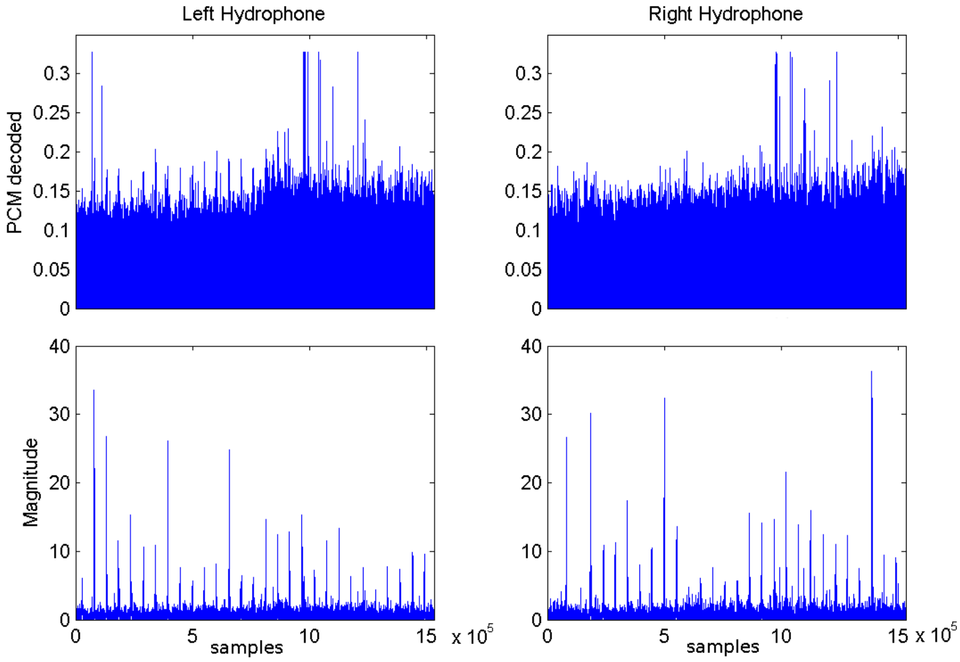


Fig. 9: Frequency analysis of the stereo hydrophone signal in presence of noise. The top plots show the raw signal, the bottom plots show the signal magnitude at the target frequency as estimated by the sliding Goertzel algorithm.

where θ_{IMU} is the change in orientation relative to the last update. The mean-free random Gaussian noise vector $\mathcal{N}(0, \Sigma)$ represents the expected movement between two consecutive sonar readings.

The linear velocity is updated using the thruster control values. Let R_θ be a rotation matrix representing the current heading and μ the linear velocity vector of a particle. Let l and r further be the control values for the left and right thruster. Assuming a pure forward motion of the robot, i.e. no side-ways drift is present, the change in velocity can be estimated as

$$\frac{d\nu}{dt} = \beta \cdot \left(\alpha \cdot R_\theta (l + r, 0)^T - \nu\right), \tag{4}$$

where α represents the speed of the robot relative to the thruster speed and β represents the acceleration speed of the robot. Both α and β are determined empirically.

c) Importance Weighting: Particles are weighted and resampled each time a new wall feature was detected. The new weight is estimated as

$$w_{t+1}^{[m]} = \lambda + (1 - \lambda)\hat{w}_{t+1}^{[m]}. \tag{5}$$

Here, the value λ is used to limit the effect of false positive wall features. The base weight $\hat{w}$ is defined as

$$\hat{w}_{t+1}^{[m]} = \exp\left(-0.5 \cdot \frac{d(p_t, s_{t+1}^{[m]})}{\sigma^2}\right), \tag{6}$$

where p_t is the position of the wall feature relative to the robot. The value $d(p_t, s_{t+1}^{[m]})$ defines the squared distance of the wall feature to the nearest wall along the sonar beam in the map.

The nearest neighbor distance can be calculated beforehand for the whole map (see Fig. 7). Considering the current sonar heading, the closest wall along the sonar beam can be efficiently estimated using methods from 3D computer graphics, e.g. [4]. For HANSE, a combination of bisection search and distance field ray marching is used to find the intersection of the sonar beam with a wall in the distance map (see Fig. 8).

A detailed example of the localization algorithm is displayed in Fig. 11 at the end of this paper, showing real-world experiments in the river Wakenitz in Lübeck.

D. Pinger Detection

Our pinger detection setup generally follows last years successful attempt of the University of Girona. In order to locate and track the pinger of the ASV, HANSE is equipped with two H2a Hydrophones from Aquarian Audio, mounted to front of the frame. In the main case, the hydrophones are connected to the stereo microphone input of the notebook.

The $13kHz$ pinger signal is now detected using the sliding Goertzel algorithm [5]. Initial experiments are shown in Fig. 9. If a signal is detected in both hydrophone, i.e. the frequency amplitude exceeds a pre-defined threshold, the Time Difference of Arrival (TDOA) between both hydrophones is calculated. With the TDOA, the pinger can be located either using a

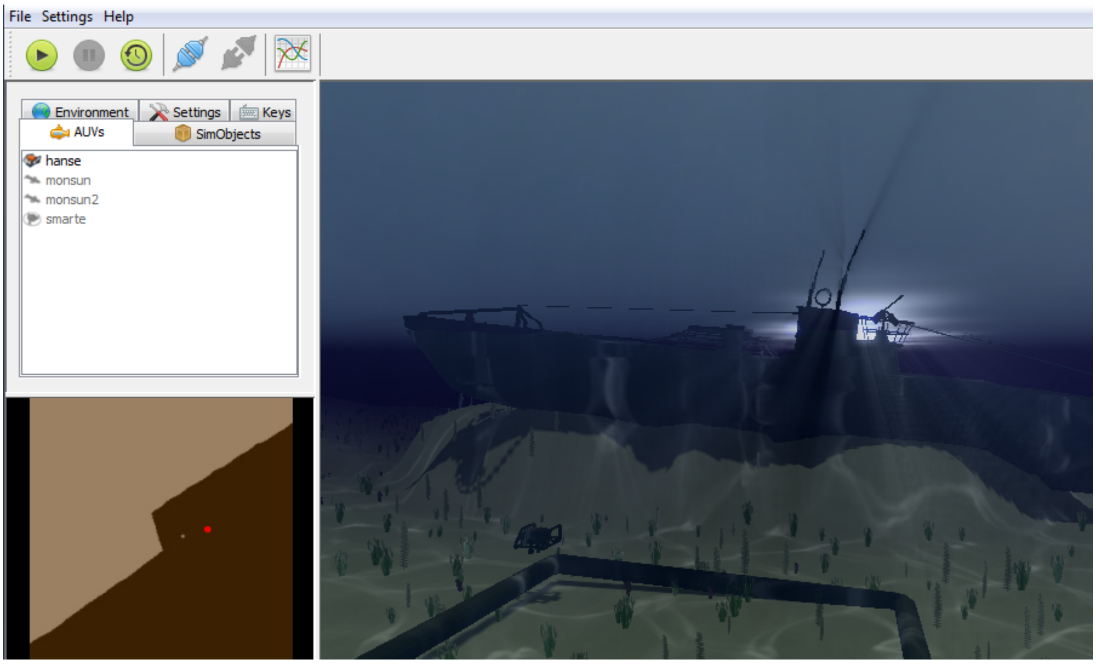


Fig. 10: Snapshot of the Marine Robotics Simulator (MARS). Showing the graphical capabilitys.

simple homing behaviour or using a particle filter-based position estimation of the pinger.

For the homing behaviour, the direction of the pinger relative to the robot can be estimated by minimizing the TDOA. Having determined the direction in which the pinger is located, the pinger may be followed by moving in the direction of increasing signal amplitude. The robot will finally surface when a given amplitude is crossed or if the maximum amplitude is reached.

Using the particle filter-based position estimation of the pinger, the TDOA defines a parabola on the water surface on which the pinger may be located. Weighting particles according to distance to the parabola, the exact position of the pinger can be estimated over time.

E. Simulation

For the SAUC-E 2011 a simulation environment was developed which proved to be a valuable tool in the development of HANSE. Hence the simulator was further improved and is now called MARS (Marine Robotics Simulator, see Fig. 10). To connect the simulator to the new ROS Framework rosjava[5] was utilized. The graphics were further improved so the camera image would look more realistic. A light-scattering effect, a caustic effect on underwater surfaces, waves and underwater plants were added. Work was also done to increase the accuracy of the physics model. Waves have now an impact on the buoyancy of the AUV. Several additional sensors and actuators were implemented and the noise capability were improved. The sonar for example features now displacement of values with different rules. For the SMART-E AUV simple servos were implemented. Also ballast tanks, simple underwater modems and a pinger detector. For a more user friendly interface the GUI was rewritten and features mouse based placement and rotation of the AUV, a map for a better overlook, context-sensitive pop-up menus and a time-based data plotter.

[5] rosjava,https://code.google.com/p/rosjava/

IV. Financial Summary

A table of the total expenses and incomes for the HANSE AUV can be found in Tab. I and Tab. II at the end of the paper, excluding travelling costs for the competitions. It lists the expenses of the last three SAUC-E combined and the expenses for the SAUC-E 2012. Total new expenses are at 1,004 Euro, where the new Hydrophones and the sponsored camera account for the 754 Euro. Total expenses are now at 14,847 Euro for the last four years.

V. Risk Assessment

Potential risks and precautions taken are listed in Tab. III at the end of the paper. In comparison to SAUC-E 2011, the cutting mechanism was removed.

VI. Conclusion

This section will provide some final conclusions on the state of the HANSE project as well as an evaluation of the work done between the last and the current SAUC-E. Sec. II introduced the AUV HANSE and gave an overview of the hardware and software components. Sec. III presented this year's innovations, where focus lied on the migration to the Robot Operating System (ROS), including the use of the state machine framework SMACH for behaviours.

An improved localization algorithm, which allows to incorporate single sonar beams, was developed and the simulation environment was extended. Finally, a new pinger detection algorithm using stereo hydrophones and a sliding Goertzel frequency analysis was introduced.

A first success in terms of the new software framework could be achieved with the launch of the second Lübeck team SMART-E, which benefits from the re-usability in many parts from software that was developed for HANSE. Initial experiments with the improved localization algorithm and new SMACH-based behaviours are promising. Here, especially the navigation behaviour profits from the increased update rate of the localization algorithm. With the new pinger detection algorithm, HANSE will further be able to attempt the pinger detection task with the new stereo hydrophone setup.

REFERENCES

[1] N. Otsu, "A Threshold Selection Method from Gray-level Histograms," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.

[2] S. O. Belkasim, M. Shridhar, and M. Ahmadi, "Pattern recognition with moment invariants: A comparative study and new results," *Pattern Recognition*, vol. 24, no. 12, pp. 1117–1138, 1991.

[3] D. Forouher, J. Hartmann, M. Litza, and E. Maehle, "Sonar-based fastslam in an underwater environment using walls as features," *ICAR 2011*, 2011.

[4] J. C. Hart, D. J. Sandin, and L. H. Kauffman, "Tracing deterministic 3-d fractals," *Computer Graphics*, vol. 23, pp. 289–296, 1989.

[5] E. Jacobsen and R. Lyons, "The sliding DFT," *IEEE Signal Processing Magazine*, vol. 20, no. 2, pp. 74–80, 2003.

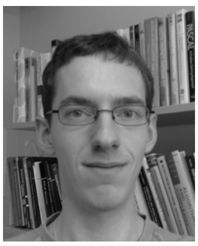
VII. HANSE TEAM MEMBERS

Jan Hartmann Project team leader. Jan Hartmann is working as a Ph.D student at the Institute of Computer Engineering, University of Lübeck. He is also a member of the HANSE team since its foundation. His interests lie in image processing and machine learning. For SAUC-E 2012, he is coordinating the software development.

Thomas Tosik Project team leader. Thomas Tosik is working as a Ph.D student at the Institute of Computer Engineering, University of Lübeck. He is part of the HANSE team since its foundation. In the last years he developed a simulation environment for AUVs. The simulator is heavily used in the HANSE development and testing.

Patrik Stahl Software engineer. Patrik Stahl is in the process of making his B.Sc. in computer science. He is a member of the team since SAUC-E 2011. His responsibilities are the implementation of the pinger behaviour and the pinger processing together with Patrick Zenker.

Patrick Zenker Software engineer and hobby photographer. Patrick Zenker is currently in the process of completing his B.Sc. in computer science. He is a member of the team since SAUC-E 2011. His responsibilities and interests are located in the implementation of the pinger behaviour and pinger processing together with Patrik Stahl. Through his hobby, photos and videos are generated at every opportunity.



Peter Hegen is a Master student at the University of Lübeck and Linköpings University. He has worked on drivers for the hardware. His interests include the interaction between hardware and software, autonomous robots as well as machine learning.

Jan Fechner Software engineer. Currently in the process of completing his B.Sc. in computer science. He is a member of the team since the last winter term. He worked on porting existing source code of the wall following behaviour and navigation using SMACH library for state handling and integrated it into the new ROS-based environment.

Cedric Isokeit Software Engineer. He is part of the HANSE Team since the last winter term and is currently in the process of making his B.Sc. in computer science. His interests are robot behaviours and the programming of embedded systems. His responsibilities in the HANSE team are the low level drivers and the management of the behaviours.

Jannis Harder Software engineer. Jannis Harder is a B.Sc. student at the University of Lübeck. He has joined the team in the last winter term to help with the transition to the ROS framework. Since then he worked on improving the scanning sonar based localization.

Sven Friedrichs Software engineer. He has joined the HANSE team in the last winter term according to his study in computer science. Sven basically worked on realizing the engine control using PID-controllers and implementing the support of an IMU.

Also involved: Hans-Joachim Reddecker, Andrea Kampsen

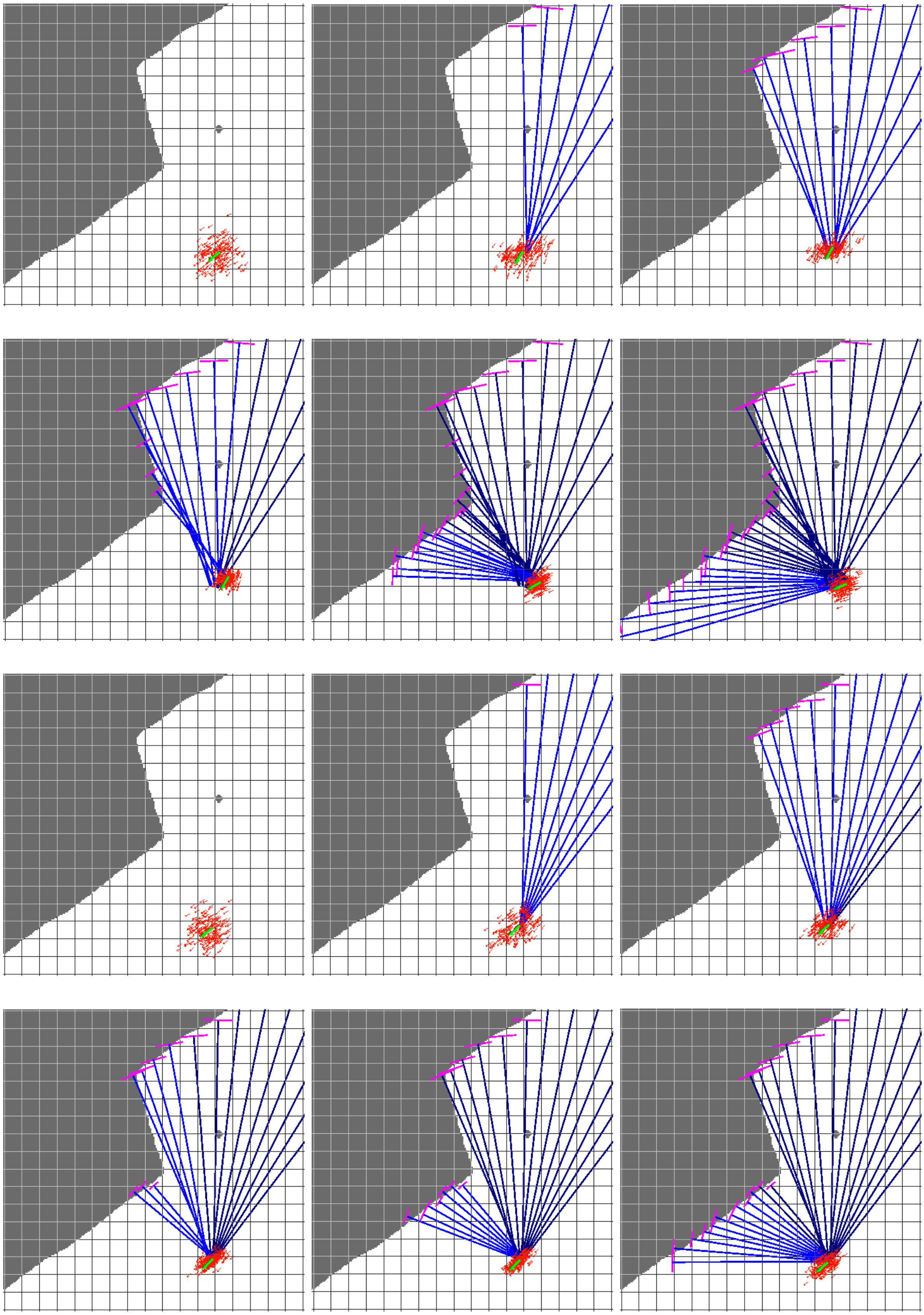


Fig. 11: Localization experiments in the Wakenitz river in Lübeck. Walls are shown in gray, particles in red, wall features in purple, and the corresponding sonar heading in blue. The mean particle position is displayed in green. Grid cells are the size of $1m^2$. The particles cloud spreads while no walls are observed (e.g. first and third row, left). It collapses when new walls are observed thus the position becomes more accurate (e.g. second row center). When wall features are distributed in a single dimension, the particle cloud elongated along the wall (e.g. bottom row, center).

Year	Item	Cost (Euro)
2009/10/11	Mechanical (base frame, case, connectors)	355
	Electrical (fuses, emergency switch, cables, etc.)	50
	4x SeaBotix BTD150 Thrusters	1,280
	4x MD22 Motorcontrollers	280
	2x 11.1V 10Ah Lithium Polymer Cells	260
	4x 11.1V 740mAh Lithium Polymer Cells	80
	2x Acer 12,1" notebooks	1,000
	Honeywell HMC6343 compass	100
	Analog Devices ADIS16354 IMU	250
	2x Intersema MS5541-CM pressure sensor and casing	28
	Imagenex Model 852 Scanning Sonar	6,000
	Imagenex Model 852 Echo Sounder	2,300
	XSens MTi AHRS	1,400
	Improved USB cables and hubs	60
	Improved WLAN stick and antennas	50
	Pinger filter chain	30
	Spare case	
	Peli case	130
	Buccaneer connectors	140
	Electrical (fuses, emergency switch, cables, etc.)	50
	total	13,843
2012	2x Aquarian H2a Hydrophones	373
	2x CrumbBoards	45
	2x WLAN adapter	43
	4x 11.1V 1000mAh Lithium Polymer Cells	112
	Electrical (fuses, emergency switch, cables, etc.)	50
	Sponsored	
	Camera DFK 22AUC03 + lens	381
	total	1,004
	sum	14,847

TABLE I: Total expenses for the HANSE AUV. Expenses for SAUC-E 2009, 2010 and 2011 are listed on top, this year's expenses below. All costs are listed in Euro after taxes.

Year	Item	Cost (Euro)
2009/10/11	innovation award SAUC-E 2009	2256
	best-recovery award SAUC-E 2010	1000
	first prize SAUC-E 2011	4,000
	total	7,256
2012	none	0
	total	0
	sum	7,256

TABLE II: Total income for the HANSE AUV. Income for SAUC-E 2009, 2010 and 2011 are listed on top, this year's income below. All incomes are listed in Euro.

Risk	Precaution
Loss of control	• timers end tasks if a behaviour fails • thruster speed will be set to 0 on communication failure • kill switch
AUV Recovery	• AUV will surface if thrusters are off • AUV can be easily recovered using e.g.
Collisions with objects or wall	• AUV speed too low to damage AUV in collision
Injuries due to lifting the AUV	• low weight • AUV can be easily lifted by two people at any point of the base frame
Injuries due to sharp edges	• most parts on the robot made of plastic • no sharp edges on other exposed parts (frame and case)
Injuries due to thrusters	• propeller casing secures the thrusters
Injuries due to electrical shocks	• low voltages and currents • fuses in all main power lines in the case

TABLE III: Risk assessment for the HANSE AUV.