RoboCup Rescue 2015 - Robot League Team PANDORA (Greece)

Loukas Petrou, Andreas Symeonidis,
Charalampos Dimoulas, Emmanouil Tsardoulias

Department of Electrical and Computer Engineering
Aristotle University of Thessaloniki
Aristotle University Campus
541 24 Thessaloniki, Greece
http://pandora.ee.auth.gr

Correspondence email: loukas@eng.auth.gr

Abstract. Within the context of the 2015 RoboCup-Rescue competition (www.robocup.org) the PANDORA Robotics Team of the Aristotle University of Thessaloniki has developed an experimental robotic platform dedicated to exploration and victim identification. Our robot is able to autonomously navigate through unknown space (e.g. building ruins after an earthquake), avoid obstacles, search for signs of life and identify victims. We are going to use a 4-wheel drive robotic platform aiming at identifying victims residing in the yellow arena and the radio drop-off zone. The current document is the TDP of the PANDORA robot.

Introduction

The PANDORA Robotics Team (**P**rogram for the **A**dvancement of **N**on **D**irected **O**perating **R**obotic **A**gents) of the Department of Electrical and Computer Engineering (DECE) of Aristotle University of Thessaloniki (AUTH), Greece aims in developing an experimental robotic platform for indoor exploration and victim identification. Overall objectives of the team are the application of the existing know-how on a real-life problem and the advancement of the group's state-of-the-art expertise. The PANDORA Robotics Team was founded in 2005 and has already participated in the RoboCup-Rescue 2008, 2009, 2011 and 2013 competitions. This year, the team intents to participate in the yellow arena and the radio drop-off zone.

The major changes compared to the previous components and systems used, are: a) a 4-wheeled platform with a simpler arm is used, b) 3D SLAM and navigation is employed, c) a set of new vision algorithms is applied, d) an algorithm for LandoltC detection has been added, e) an upgraded mainboard and a dedicated DSP board are installed. Additionally a number of new

sensors are installed: an Xtion Pro RGB-D sensor, a Flir thermal camera an Attitude and Heading Reference System device, a Leddar Range Finder, a 20m Laser Range Finder and a set of cameras.

1. Team Members and Their Contributions

The team comprises 3 faculty members of complementary expertise, one post-doctoral researcher and a compilation of postgraduate and undergraduate students. The following list provides the names and responsibilities of the team members.

Team Mentors

- Loukas Petrou, Associate Professor
- Andreas Symeonidis, Assistant Professor
- Charalampos Dimoulas, Assistant Professor
- Emmanouil Tsardoulias, Post-doctoral researcher

AI Team

Team Leader: Emmanouil Tsardoulias
SLAM: Konstantinos Samaras Tsakiris
Navigation: Dimitrios Kirtsios
Data Fusion: Konstantinos Mavrodis
Decision Making: Konstantinos Chamzas
Planner: Dimitrios Kirtsios

Software Architecture Team

Team Leaders: Christos Zalidis, Christos Tsirigotis
Testing: Konstantinos Sideris, Eirini Chatzieleytheriou
Simulation: Dimitrios Geromichalos, Konstantinos Zisis
Integration: Giannis Lykartsis, Kostas Peppas, Spyridon Papatzelos

Vision Team

Team Leader: Marios Protopapas
Victim recognition: Vasilis Bosdelekidis, Aggelos Triantafyllidis
Tag/Hole/Motion detection: Alexandros Filotheou, Vasilis Choutas
QR – Landolt C recognition: Miltiadis Kofinas

Sound detection Team

Sound control: Nikolaos Tsipas, Eleutherios Mylonakis

Electronic design Team

Team leader: Konstantinos Panayiotou
Sensors: Nikolaos Taras
Motors: Elisavet Papadopoulou
Integration: Georgios Kouros
Arm and Servos Design: Petros Evaggelakos, Alexios Papazoglou

Mechanical design Team

Team leader: Nikolaos Michailidis

The team is going to be represented by approximately 10 members in the competition. Names are going to be listed in the registration form.

2. Operator Station Set-up and Break-Down (10 minutes)

Three operators are needed for setting up the PANDORA robot: the head operator of the system who carries the base station case, accompanied by two operators that carry the platform case.

The initialization process is realized as follows:

- Transfer all objects in the area and deploy (3 minutes).
- Activate the platform and the base station (3 minutes).
- Launch the PANDORA robot OS (2 minutes).
- Perform communication check, in order to establish and validate Wi-Fi connection (1 minute).
- Perform system check and diagnostics, in order to verify that all the components of the platform are working properly (1 minute).

3. Communications

Following RCR regulations, we are going to use W-LAN 802.11a (5 GHz) and will wait to be assigned with a channel/band from the organizers during the competition.

Table 1. PANDORA communication protocol

Rescue Robot League		
PANDORA (GREECE)		
Frequency	**Channel/Band**	**Power (mW)**
5.0 GHz - 802.11a		100

4. Control Method and Human-Robot Interface

The PANDORA robot will operate in two modes: the *fully autonomous mode*, where a number of concurrent processes will be deployed in order to achieve autonomous exploration and victim identification and the *tele-operation mode*, where the robot will be totally manipulated by an experienced user.

In order to ensure a flexible and modular scheme where reconfiguration is possible, we opted for a component-based software architecture. The selected architecture ensures easy testing and integration.

4.1 PANDORA Software Architecture

It is a common knowledge that autonomous robots are highly complex systems that require both HW/SW integration, as well as the deployment of multiple heterogeneous modules. Middleware frameworks aim to minimize this complexity by providing infrastructure and tools for building modifiable and reusable solutions, while successfully dealing with communication issues, component heterogeneity, coordination, task scheduling and allocation [1]. Apart from the above, middleware added-value relates to major non-functional requirements they ensure, such as real-time (or near real-time) performance, reliability and security. From the plethora of existing approaches, though, not many satisfy the above criteria.

Having considered various off-the-shelf middleware (including MSRS, OROCOS and ROS), we adopted ROS[1] for PANDORA's middleware. A number of factors were considered during the middleware selection process. A messaging communication scheme was preferred to a typical RPC-style middleware, due to its inherent ability to promote loose coupling. Furthermore, messaging provides asynchronous communication with the ability to control dataflow, an extremely important feature for complex interconnected systems. Among others, the basic advantages of ROS are its open-source nature, transparent architecture, wide-spread usage, interoperability with other robot frameworks, quality of the development tool chain and extensive documentation.

ROS comprises a peer-to-peer network of components (denoted as nodes), communicating via messages through the respective ROS infrastructure. The channels that messages are sent through are called topics. RPC-style communication is also achieved through services and data persistence is achieved through the Parameter Server.

To achieve maximum decoupling, a modular approach was followed, thus defining various levels of abstraction. Interfaces realizing communication between components are encapsulated and decoupled from the implementation, thus providing domain-specific functionality only at a component level [2]. Functionality is logically grouped and satisfied by different packages implementing nodes that perform different tasks. The adopted software design decouples nodes from each other as much as possible, thus minimizing the induced interconnection complexity.

Figure 1 depicts the robot software architecture, where one may identify, apart from the functional modules responsible for vision, navigation, SLAM and motor control, an Agent module that orchestrates robot actions and defines the robot's strategy. The fact that the robot integrates various controllers, sensors and actuators, thus handling different types of data, led us to the establishment of a data fusion layer, responsible for the system's overall

[1] http://www.ros.org

health control and management. This layer aggregates low-level sensor measurements concerning victim detection and identification and is responsible for performing sensor fusion, filtering and forwarding values which may correspond to candidate victim locations or directions. This way, we succeeded in substantially reducing the information overhead, a key factor for the efficiency of the robotic system.
The basic modules developed are:

- ***Hardware Interface***: it controls data flow from and to the microcontrollers, acting as a Hardware Abstraction Layer for the robot. This module handles most of the sensors, such as thermal and voice sensors. Also, it handles motor controllers, responsible for the vehicle's motion.
- ***SLAM***: The SLAM module is responsible for performing Simultaneous Localization and Mapping, as well as storing and providing the map data. It is based on measurements received as laser scans by the laser's driver.
- ***Control***: solves the vehicle's kinematic model and produces linear and rotational velocities enabling the vehicle to move. Moreover, it calculates the desired motion of each actuator, such as servos that orient sensors. This module is responsible only for calculations and does not interface with the actual hardware, achieving a hardware agnostic abstraction.
- ***Sensor Preprocessing***: handles raw data received from thermal, sound and CO2 sensors. It uses pattern recognition algorithms in order to produce higher-level data representations, used later to construct a world model.
- ***Vision***: responsible for handling the cameras used for exploration and finding possible victim locations, as well as the camera used for detailed victim identification. Handles RGB images as well as RGB-D data produced by depth sensors.
- ***Navigation***: responsible for motion planning and navigating the robot through the unexplored regions of the map or towards a possible victim for further identification.
- ***Data Fusion***: it decouples low-level sensor measurements for victim identification and high-level navigation components. Furthermore, it is responsible for performing sensor fusion, constructing a model of the perceived world and forwarding possible victim positions (or directions) to the *Agent*.
- ***Agent***: it performs higher-level decision making and is responsible for orchestrating actions performed by all other modules by defining their state. Practically, *Agent* defines the strategy to be followed during the course of the competition.
- ***GUI***: provides a Graphical User Interface for the robot operator. In addition it provides the remote control features for tele-operation.

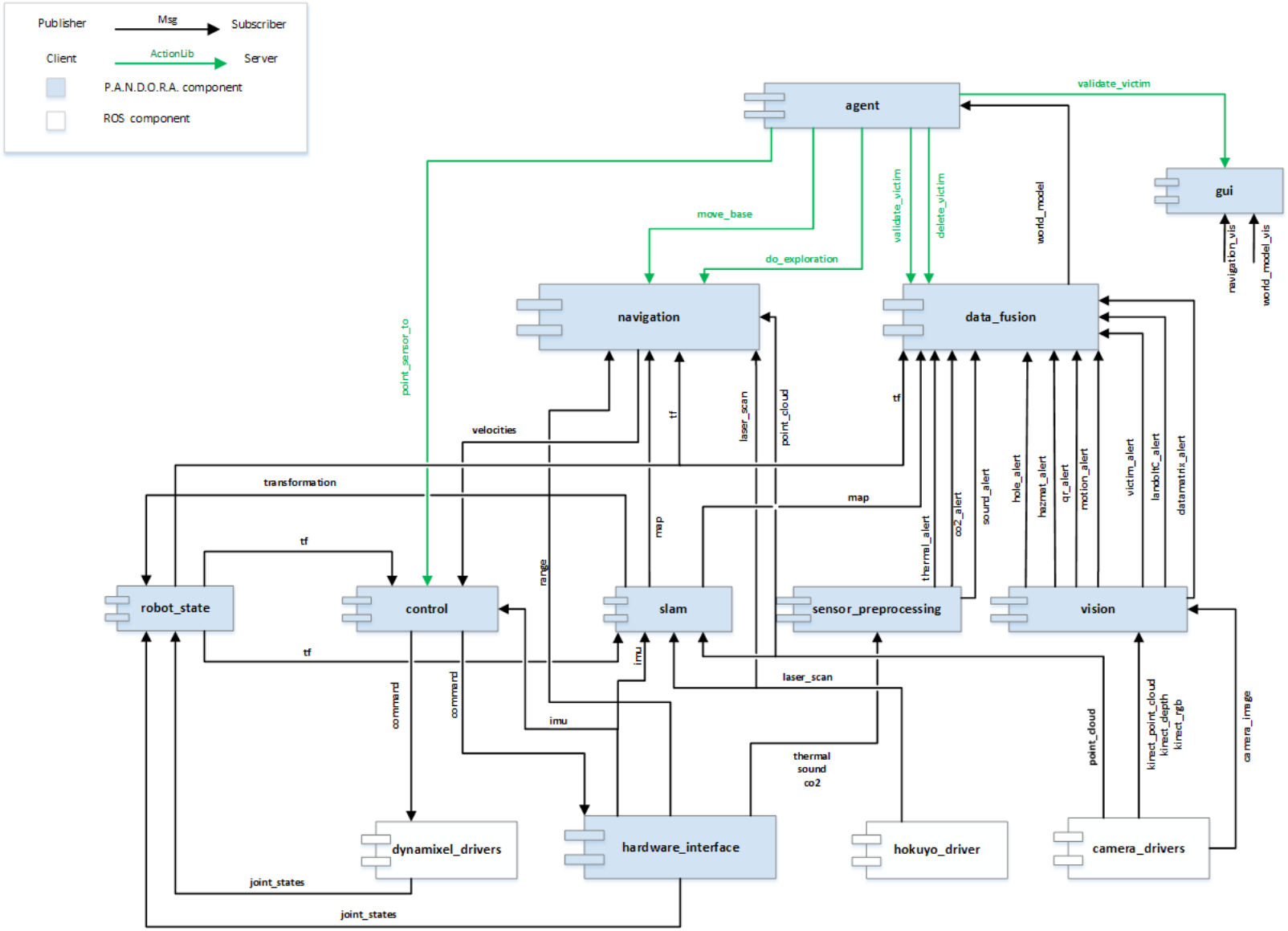

Fig. 1. PANDORA component diagram. Communications represented as ROS messages and actions.

4.2 PANDORA Graphical User Interface (GUI)

PANDORA provides a user friendly GUI for visualizing information and operating the robot. Several measurements are available in real time, such as battery voltage levels, temperature and CO_2 values and sonar readings. Nevertheless, the operator can dynamically add/remove sensor information and modify the type and the layout of the widgets displayed, since PANDORA's GUI adopts a widget-like architecture. Other visual cues, such as the world map , robot trajectoryetc, are displayed through *Rviz.* Rviz is a tool that provided from ROS and can be used to display several information about the robot in real-time or to interact with it. A screenshot of the GUI is provided in Figure 2, while the Rviz GUI in Figure 3.

When on tele-operation mode, the robot vehicle is controlled using a wireless gamepad or a keyboard, while its sensor's orientation can be controlled using Rviz's *Interactive Markers.* While on the autonomous mode, GUI is used only for visualizing/monitoring purposes and no intervention is allowed, up to the point that a victim is recognized. Then, PANDORA sends an interrupt signal to activate the GUI and expects proper operator action in order to continue.

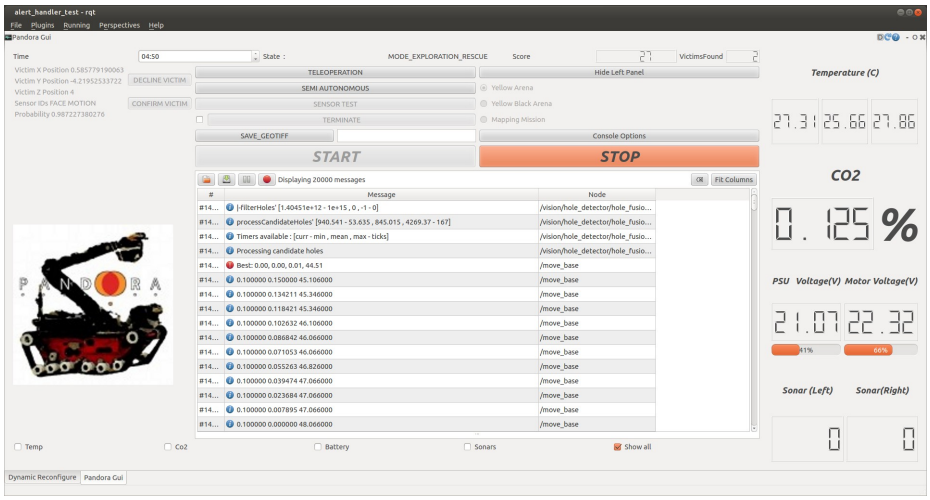

Fig. 2. PANDORA GUI – On the right pane, sensor information, on the center screen, information on victims.

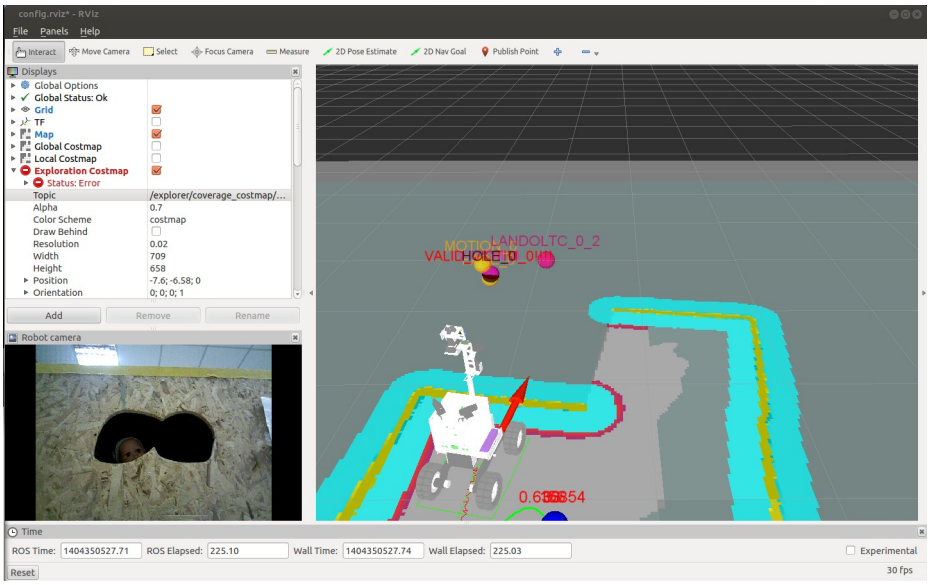

Fig. 3. Rviz GUI – Showing the perceived world model and a live stream from the robot's camera (bottom left).

4.3 PANDORA Hardware Architecture

PANDORA's system topology is based on the well-known Star-Network. In order to perform high computational power robot tasks, an Intel i7 CPU attached on a Mini-ITX motherboard (MI980 from IBase), is used as the central processing node. All the required data are stored in an SSD drive. Information data from the various sensors and to actuators is managed by the central process node.

The robot platform is equipped with two sets of sensors. The first set is responsible for localization and navigation procedures, while the second for victim identification. The current robot's hardware architecture is presented in figure 4.

The Head board controls the functions required by the set of sensors indicated in Figure 4. Those are a magnetic rotary encoder, an IR thermal camera (FLIR Lepton), a CO2 sensor and three ultrasonic range finders. The Head board is also responsible for reading the actual voltage level of the two batteries, currently installed on the platform. For these purposes a 32 bit ARM Cortex M4F microcontroller (LPC4088) is used. Communication with the central process node is achieved via an USB (serial-ACM) interface.

A set of stand-alone sensors are currently installed on the robot platform:

- Two Web cameras: Used by QR and Hazmat detection procedures.
- Hokuyo UST-20LX or alternatively Hokuyo URG-04LX: Laser Range Finder (LRF), used for mapping, localization and navigation procedures.
- Leddar Range – Distance Sensor: Used for localization and navigation procedures. Also used as auxiliary to the ultrasonic range finder sensors.
- Xtion Pro: RGB-D camera used for identification procedures. Can also be used for localization and navigation.
- PNI Sensor Trax AHRS: Inertial Measurement Unit (IMU). Used for localization and navigation procedures. This unit is also currently used for the stabilizer module.

Two Pan & Tilt modules are currently installed. The first one is used for motion control of the head of the robot. The head of the robot carries the victim identification sensors. The second one controls an RGB-D camera and the thermal camera, currently installed on the base of the platform's chassis. Furthermore a stabilization platform is employed in order to ensure leveled readings from the LRF sensor regardless of the robot's inclination.

The main motion of the robot is achieved by four MAXON brushless DC motors, one for each wheel. Each motor is connected on an EPOS2 module motor controller. The motor controllers are all connected on a CANBus, thus each motor can be controlled separately. The central process node communicates with the motor control devices attached on the CANBus, via an RS232 interface.

In order to enable easy debugging of the hardware components and their intercommunication, a detachable module with an embedded microcontroller and an LCD screen can be connected to the Head board microcontroller and probe the system for correct functionality and possible errors.

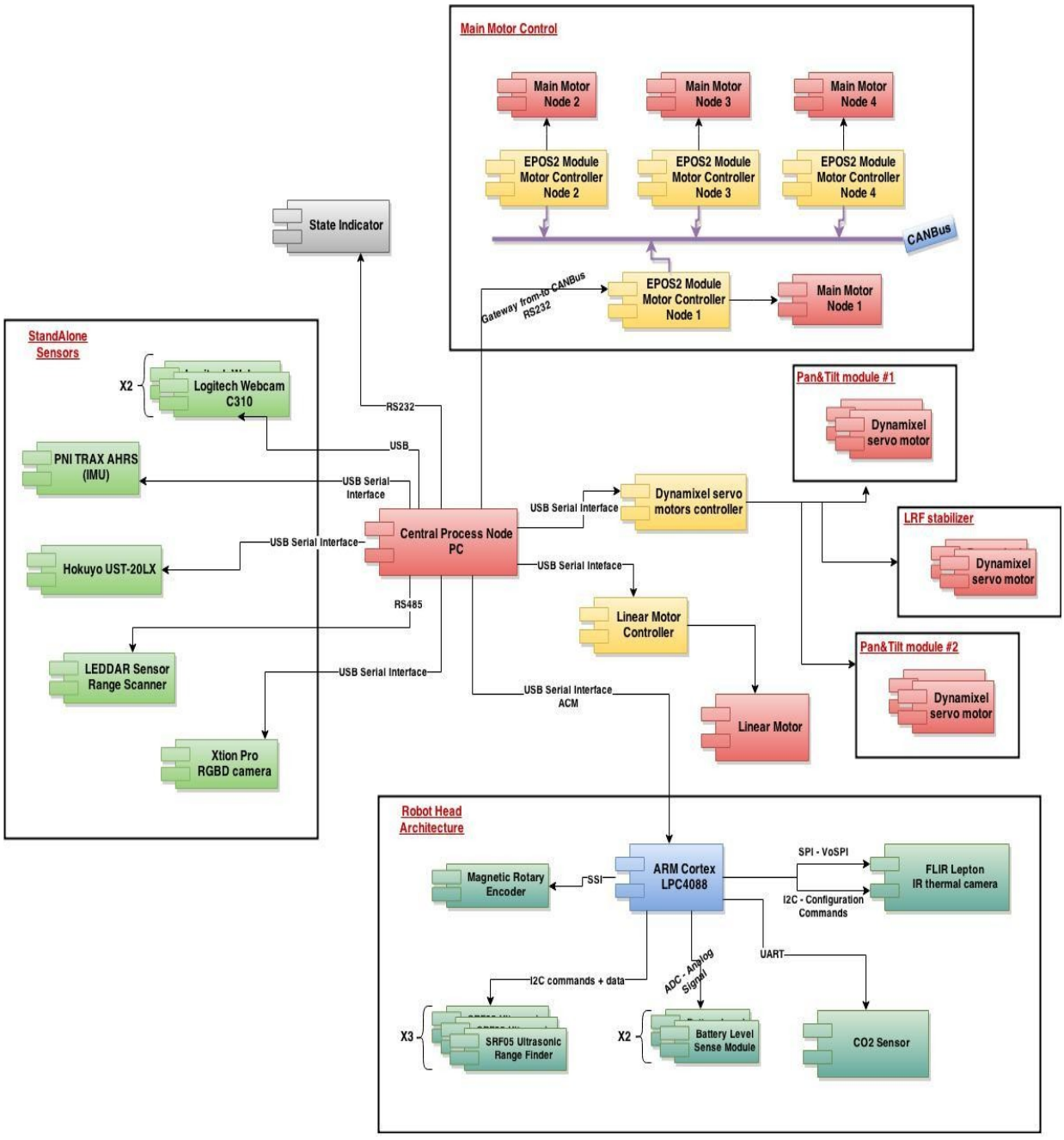

Fig. 4 PANDORA hardware architecture

5. Map generation/printing

5.1 SLAM

The 2D mapping algorithm used by the PANDORA team is CRSM SLAM [3], [4], where CRSM stands for Critical Rays Scan Match. CRSM SLAM relies solely on the Hokuyo LRF sensor. A metric map and specifically an occupancy grid map is generated, where each cell holds the probability to be occupied.

CRSM SLAM comprises two key features: the ray selection and the scan matching procedures.

- **Ray selection:** In order to reduce the execution time and computational requirements of the algorithm, a ray-picking method is employed. Specifically the only rays participating in the scan match procedure are the "critical" ones. A ray is denoted as "critical" if it provides additional information in comparison to the mean information gain extracted from the laser scan, or in other words if the specific ray can be described as a feature of the scan. Ray selection is performed via heuristic methods which include extracting the scan parts and picking rays accordingly to local ray density.
- **Scan matching:** The rays picked from the previous step are used for the scan matching procedure. The method employed is a hill climbing algorithm, and specifically RRHC (Random Restart Hill Climbing) [5]. Scan matching is deployed to calculate the geometrical transformation between a laser scan and its predecessor. RRHC is a simplified form of a genetic algorithm that involves just one individual. In CRSM SLAM's case the RRHC genome is G = [D_x, D_y, D_{theta}], i.e. the geometrical transformation, where D_x, D_y, D_{theta} are the changes in the robot's x,y and angle coordinates. RRHC finds the most accurate G that better aligns the current scan with the global map.

Once the correct geometric transformation is acquired, the occupancy grid map as well as the robot's pose is updated. The algorithm is obviously iterative and the typical iteration frequency is 10Hz, of utmost importance, as the computations must be performed at real time. The result of the CRSM SLAM is shown in Figure 5.

CRSM SLAM has the disadvantage of providing a two dimensional space representation, something that is rather limiting for advanced autonomous navigation capabilities. Thus, a 3D variant of CRSM SLAM is created. In order to do so, a depth providing sensor is required, which in our case is the Xtion Pro platform. The measurements required consist of a point cloud including the distances of the environment in a grid formulation of 640 to 480 pixels.

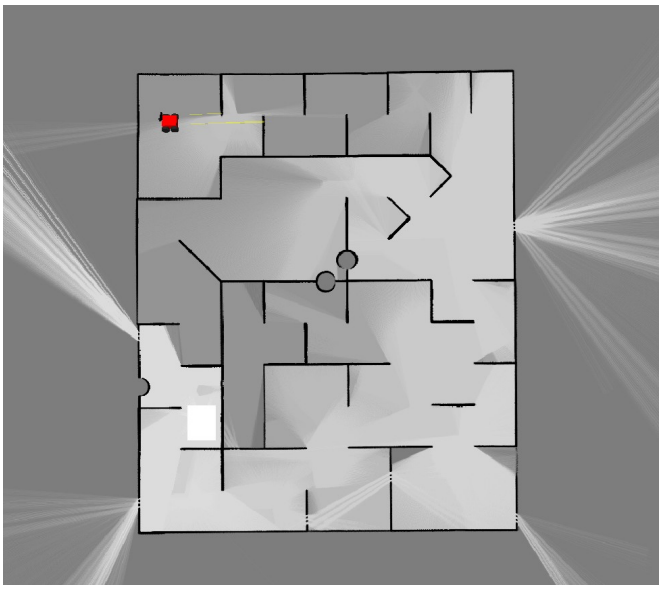

Fig. 5: CRSM SLAM result

Since the direct employment of the entire point cloud in a scan matching algorithm is impossible to be performed in an – almost – real time manner, a similar heuristic ray picking method to the CRSM SLAM will be employed. This time, the selected entities that represent the 3D scan's features, may not be just points, but even more advanced ensembles of rays, like lines or corners.

Finally, a dimension reduction technique will be used in the 3D scan match that derives directly by the conjunction of the initial CRSM SLAM with a 3D mapping algorithm. Since CRSM SLAM provides accurate enough measurements about the [x, y, yaw] coordinates of the robot, the 3D scan matching algorithm take these as input, and reduces the computational requirements by searching in the [z, roll, pitch] dimensional space.

The produced 3D map will have the form of an Octomap [6] and will be employed from the autonomous navigation module to perform path optimization techniques, in order to ensure safe and efficient robot navigation.

Finally, the 2D map produced by the CRSM SLAM or a horizontal slice of the product of the 3D slam, will be transformed to a Geotiff form.

5.2 Navigation

PANDORA's navigation module combines state of the art techniques and algorithms, regarding path planning in 3D cluttered environments. Our path planning approach is oriented in environment exploration and full space coverage. As the main goal of the mission is to identify victims, as "covered" is denoted the space investigated by sensors responsible for victim identification. Navigation module comprises four sub-modules: the *Global Planner,* the *Local Planner,* the *Target Selector* and the *Navigation Controller.*

- ***Global Planner*** is responsible for generating a path, given a goal in space, without taking into consideration the kinematic constraints of the vehicle (holonomic constraints). We use OMPL (Open Motion Planning Library) [7], to solve path planning in 3D, using an efficient octree-based [8] representation of the 3D world. OMPL takes care of the computationally intensive task of collision checking between the mobile platform and obstacles, using FCL (Flexible Collision Library) [9].

- ***Local Planner*** is responsible for generating a trajectory, given a path as input (previously generated by the *Global Planner),* with respect to the kinematic constraints of the vehicle. The *Local Planner* performs 3D collision checking with obstacles using FCL, but the search for commands controlling the translational and rotational velocity of the robot is carried out directly in the space of velocities. This is done by reducing the search space to a *dynamic window* [10], which consists of the velocities reachable within a short time interval.

- ***Target Selector*** is responsible for selecting the next goal that the robot will move to. It uses heuristic and probabilistic approaches to optimise an objective function. *Target Selector* takes as input the 3D representation of the world and a 3D coverage patch dictating the amount of the environment covered by sensors responsible for victim identification, both in Octomap representation. The output goal can be an approach point of a possible victim (if such a victim has been identified by other modules) or a point in space which serves best the particular exploration strategy.

- ***Navigation Controller*** plays a critical role, as a coordinator of all the above sub-modules. *Navigation Controller* distributes all the necessary information between the different sub-modules and coordinates the communication between them. Additionally, it is responsible for recovery behaviours in case the robot is incapable of moving (stuck situation) or doesn't have enough information to proceed with target selection and path planning.

5.3. Data fusion

The Data Fusion module is responsible for filtering out the messages generated by PANDORA sensors (CO_2, Thermal camera, sound module). It stores thresholds of all sensors, denoting the probability values of an eligible valid measurement. Given that a sensor measurement exceeds the threshold, Data

Fusion informs the Agent, also providing details. One should mention that thresholds are not hard-coded and are not crisp; an elaborate data fitting scheme has been introduced in order to tune raw sensor input before accepting them. Furthermore, based on the proper modeling of data sources, sophisticated machine learning mechanisms for anomaly detection are applied in order to exclude any abnormal behavior of sensors. Messages communicated by Data Fusion abide by a predefined uniform format, containing the sensor type, the probability of a measurement to be valid, and its direction, in case the sensor is directional.

Apart from its basic functionality, Data Fusion also blends low-level sensor measurements and high-level data from Navigation and Vision in order to create an advanced strategy for victim identification. Probabilistic, density-based and kernel machine approaches will be considered in order to build a robust data fusion mechanism.

6. Sensors for Navigation and Localization

The PANDORA robotic platform is equipped with several sensors in order to determine its current position and its distance from various objects. These sensors are discussed next.

6.1 Laser Range Finder

For map creation the Hokuyo URG-04LX[2] Laser Range Finder (Figure 6) has been installed. It has a viewing angle of 240° and a detection range of 20mm up to 4m. The angular resolution is 0.36°, which gives 667 measurements in a single scan, while its linear resolution is 1mm. Measurement accuracy varies from 10mm (for distances from 20mm to 1m) to 1% of the measurement for distances up to 4m (Figure 6). It operates on 5V DC (possible error of +/- 5%) and has a current consumption of 500mA.

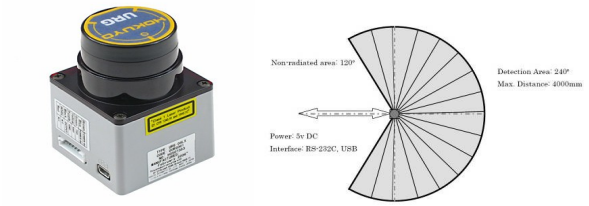

Fig. 6. Laser sensor (Hokuyo URG-04LX) and its field of view

For longer distances the Hokuyo UST-20LX[3] Laser Range Finder (Figure 7) can be alternatively used. It has a viewing angle of 270-degrees and a detection range of 20mm up to 20m. The angular resolution is 0.25-degrees, which gives 1080 measurements in a single scan. Positions of objects in the

[2] https://www.hokuyo-aut.jp/02sensor/07scanner/urg_04lx.html
[3] http://www.hokuyo-aut.jp/02sensor/07scanner/ust_10lx_20lx.html

range are calculated with step angle and distance. It operates at 12V (or 24V) DC and has a current consumption of 150mA.

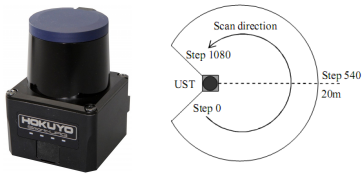

Fig. 7. Laser sensor (Hokuyo UST-20LX) and its field of view

6.2 Ultrasonic Sensors

Three ultrasonic SRF05[4] sensors (Figure 8) are situated around the robot. They communicate via the I2C bus with the ARM CortexM4F microprocessor, publishing a pulse with width proportional to the distance of the object. Their power consumption is very low (approx. 0.02W). In the front part of the vehicle they will be used to prevent the vehicle from bumping on obstacles. SRF05 sensors have a detection range of 3cm to 4m and will be used as a complement to the Laser Sensor.

Fig. 8. Devantech SRF05 Ultra Sonic Ranger

6.3 Leddar Range Finder

In order to perform z axis estimation, as a part of the SLAM algorithm, the LEDDAR M16[5] sensor module is installed (Figure 9). The Leddar M16 Sensor Module is an advanced sensing solution that combines 16 independent active elements into a single sensor, resulting in rapid, continuous and accurate detection. It has a 45 degree horizontal field of view and a detection range of 0 to 100 meters, while its linear resolution is at 10mm. Measurement accuracy is 5cm at a range of 100 meters. It operates at 12V DC and has a power consumption of 4W. Communication with the host device is achieved via RS485 protocol.

---

4 http://www.robot-electronics.co.uk/htm/srf05tech.htm

5 http://leddartech.com/en/products/leddar-m16-module

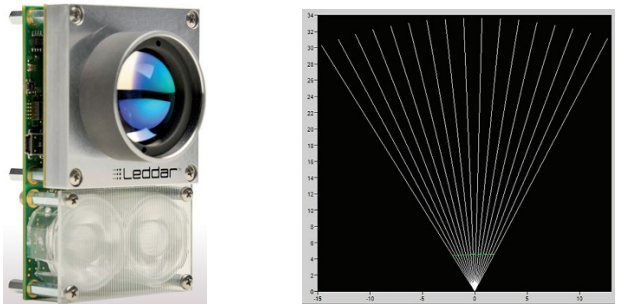

Fig. 9 The Leddar M16 sensor and its field of view

6.4 RGB-D sensor

As RGB-D sensor, the ASUS Xtion Pro live[6] (Figure 10) is used. It features an RGB camera and an IR depth imaging sensor. In this manner an RGB-D video, a depth image and a point cloud are produced. It exhibits an angular Field-of-View of 57 and 43 degrees, horizontally and vertically, respectively, with a frame rate of 30 Hz, a spatial range of 640 x 480 and a nominal depth resolution of 1 cm in 2 m distance. In the context of PANDORA, the RGB-D sensor is used primarily in computer vision and SLAM. SLAM uses the output point cloud to construct a 3D map, whilst the depth and RGB images are used in PANDORA's computer vision module in order to detect certain visual key features, such as QR codes and possible victim places.

Fig. 10. The utilized RGB-D camera: ASUS Xtion Pro live

7. Sensors for Victim Identification

A number of different sensors have been installed and are utilized providing input to sophisticated detection algorithms, in order to accurately identify a victim and pinpoint his/her location. Specifically, a multimodal vision system, thermal sensors, a CO2 sensor and two different microphone arrays are being used. Sensor results are then fused to determine the behavior of the robot.

---

[6] http://www.asus.com/Multimedia/Xtion_PRO_LIVE/

7.1 Vision

A multimodal vision system has been implemented in order to enhance the various aspects of the victim identification task. The core of the system is an RGB-D camera a thermal camera and two (or probably more) web cameras. This core module allows for both classical image processing techniques to be combined with depth-based features and imaging approaches in order to better serve the vision requirements.

During *tele-operation*, the RGB camera transmits a video stream to the control station, for the operator to have a visual sense of the robot's surroundings in real time.

All vision modules have been developed in C++ and heavily exploit OpenCV libraries [11] through ROS. PANDORA *Vision* provides the following functionality:

- *Hole detection and localization.* Detection is performed by fusing the information extracted by two separate modules:

i. Enhanced depth-based edge detection of a denoised and iteratively processed version of 2D scene image is used to detect closed boundaries, which are then considered as BLOBs. There is a sequence of complementary processing tasks, namely: a) noise elimination in terms of depth artifacts detection and restoration using multimodal parametric interpolation /filtering; b) edge detection and thresholding followed by multi-parametric binary image enhancement to serve edge denoising (i.e. edge dilation, thinning, unwanted edges removal, edges validation, spatial boundaries boosting and final pruning); c) BLOB detection and validation; d) checking and final recognition /exporting of detected holes.

ii. Edge detection of a denoised version of 2D scene image is used to detect closed boundaries which are considered as BLOBs [12]. Parts of the depth-oriented methodology is utilized and adapted for the case of RGB images, in combination with standard colored image BLOB detection. The processing chain is organized in the following main phases: a) detection of areas neighboring to wall surfaces by means of color and structural segmentation of the scene image; b) extraction of dark regions that usually correspond to holes via intensity thresholding; c) hole detection enhancement, taking depth information into consideration.

Fusion of the above cues results to connected components at the locations of holes on a single image. Time persistent BLOBs are detected as holes. Size and shape constraints enhance detection accuracy, while closer view decisions might be engaged (after vision and data fusion) in order to further improve detection accuracy. Localization consists of 2D information and corresponds to the specification of the direction at which the hole is detected. Most of the above techniques, especially the ones that are involved in the depth

imaging processes, are based on newly introduced algorithms, avoiding computationally heavy point cloud detection and 3D image reconstruction procedures, thus increasing the CPU performance.

- ***Victim detection***. A superset of RGB-D-T features is utilized, including color depth and thermal – related metrics, along with their structural features and classical face recognition features /algorithms [13]. A support utility allowing training on a custom set of objects has been developed to allow for replacing real human faces by other human-like artifacts (e.g., dolls). In addition to the above, masks of the detected-BLOBs are also utilized in combination with -within BLOBs- depth and structure variation metrics to further validate the binary face (/victim) detection outcome.
- ***Motion detection***. A video motion detection module has been implemented in order to facilitate the victim identification task. The current system is triggered by a high possibility BLOB detection message and it operates while the robot is not moving. Frame and multi-frame differencing are adopted as in case of [14], in combination with adaptive foreground / background segmentation, by means of iterative exponential averaging in time-space or wavelet domain [14] - [15]. The last algorithm initiates taking the first frame as the background image, and proceeds by iteratively updating the background by exponential averaging past values with the new intensities of the non-moving pixels. A threshold, which is also iteratively updated through exponential averaging, is used to separate moving pixels from still regions. The extracted motion is estimated as the absolute difference between the frame intensity and the background image, in a per-frame basis. This method is relatively fast and computationally inexpensive, while its adaptive nature allows for fast adaptation and satisfactory performance, even in poor lighting conditions, and especially in cases that background objects are not changing. Once video motion detection is engaged, motion analysis runs for a short time period (a few seconds) quantifying the estimated motion activity. Then, by comparing with a predefined threshold, a decision is taken whether victim motion is present or not.
- ***Hazmat and Eye chart pattern detection***. A general purpose logo detection algorithm has been developed for detecting specific flat shape patterns been robust by applying affine transformations [16] - [17]. The algorithm is trained to recognize a pattern on the basis of the SIFT [18] salient point descriptors – their value and the geometry of their locations. Test images are processed for the extraction of SIFT local points and a greedy procedure is followed in order to obtain the best possible alignment of pattern's local descriptors to the (many) similar descriptors of the test image. A K-D tree is used for speeding up this procedure. Geometric compatibility combined with appearance compatibility (measured as the similarity between the descriptors) is used to determine whether the pattern appears on the examined image. The same process is repeated for each pattern.
- ***QR code detection***. For the purpose of detecting and decoding QR codes, the use of Zbar, a dedicated bar code recognition API is used. In addition, a camera selection and configuration procedure was conducted, focusing

on smaller size *QR code* detection accuracy improvements. Hence, additional high resolution and intensity monochrome cameras with wide field of view were tested and are utilized for the purpose of more demanding *QR code*, but also *Hazmat* and *Eye chart pattern* detection. For this year it was decided to equip the autonomous platform with a pair of cameras exclusively used for QR detection.

- ***Landolt-C*** *detection.* Another task that was incorporated in order to further analyze and improve the object identification and discrimination capabilities of the vision system is the *Landolt-C* recognition module. The method is based on a modified concentric circle detection algorithm [19], utilizing edge gradient direction estimation along with various searching rules and voting decision schemes. As a result, the center of the concentric /nested *Landolt-Cs* is provided, as well as their number and the orientation. The proposed method is characterized by relatively fast execution and high detection accuracy, especially in cases that high resolution camera is used.

7.2 Temperature

We consider that temperature differences in the environment could imply victims. Thus we have installed a long-wave infrared (LWIR) camera, in order to compare temperature values, find fluctuations and make an estimate of a victim's position, if one is detected. The FLIR LEPTON (Figure 11) is a LWIR camera module designed to interfere easily into native mobile-device interfaces. It captures infrared radiation input in its nominal response wavelength, from 8 to 14 microns. It has a Focal Array Plane (FPA) of, 80(h) x 60(v) active pixels, with active thermal sensitivity less than 50mK (Kelvin), 51-degrees horizontal typical field of view and 63.5-degrees diagonal.

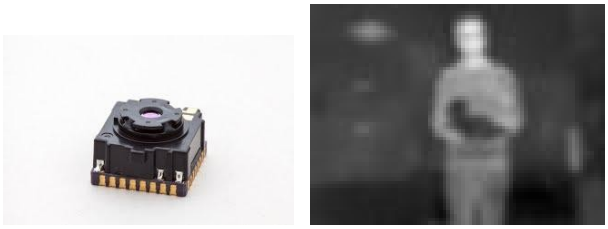

Fig. 11 The FLIR LEPTON camera and a typical image

The FLIR LEPTON LIWR[7] camera is connected on an ARM-based microcontroller. It provides a command and control interface (CCI) via a two-wire interface similar to I2C. Frame transfers are provided through the LEPTON video-over-SPI protocol (VoSPI) allowing efficient and verifiable transfer of video over a SPI channel.

[7] http://www.flir.com/cores/content/?id=66257

7.3 CO_2 sensor

The CO_2 sensor (Figure 12) measures the concentration of CO_2 gas in the environment. For the detection of the human respiration, we simply track fluctuations in the concentration of CO_2 in the air. The selected sensor can detect concentration of CO_2 gas, from 0 – 50,000ppm.

Fig. 12. DYNAMENT[8], Premier High Range Carbon Dioxide Sensor, Non-Certified Version Type MSH-P-HCO2/NC

7.4 Sound

Pandora's sound system implements two different spatial audio processing units that can be alternatively and/or collaboratively utilized in order to serve audio event detection and sound source localization.

The first module is a prototype coincident microphone array consisting of 4 miniature electret microphones with cardioid pickup, favoring the implementation of energy-based (direction of arrival) localization, along with feature-based audio event detection-segmentation. The approach was inspired by the sound-field microphone theory and the related sound source localization approaches, whereas multi-band estimation can be further deployed for noise reduction and accuracy improvements purposes [20] – [21]. The unit comprises the following components:

- Four electret-cardioid microphones (Figure 13) placed at the same level (considered to be the z=0 plane) and position, forming a coincident microphone array. Thus, the principal pick-up axes (main directivity vectors) form a cross shape, with each pair of successive microphones having an angular main axis distance of 90°. In geometrical terms, each microphone points on one of the four distinct directions $x^+(1,0,0)$, $y^+(0,1,0)$, $x^-(-1,0,0)$, $y^-(0,-1,-0)$ of the Cartesian XYZ.
- Four-channel signal conditioning circuits and analog to digital convertors (ADCs).
- The ARM Cortex M4F microcontroller

This unit is used for audio events detection (adaptive thresholding) and energy based sound source localization. Once event detection is decided, direction of arrival (DOA) localization is performed to estimate the horizontal DOA angle

[8] http://www.dynament.com/infrared-sensors/carbon-dioxide/premier-non-certified-carbon-dioxide-sensors.htm

(θ). Specifically, the four microphones are grouped regarding their pointing axis (x or y), in order for their signals to be sequentially subtracted (x^+-x^-, y^+-y^-), thus forming two coincident, figure of eight, microphones with perpendicular main axes. In this context, the magnitude of the direction of arrival angle |θ| is easily estimated from the arctan of the energy ratio of the two figure of eight microphone signals (oriented in the y and x axes). Next, the horizontal direction of arrival is decided by estimating the quadrant where audio energy maximizes, by ordering the audio energies of the four initial (cardioid) microphones. One of the benefits of the proposed solution is its simplicity and the fact that quite low sampling frequency can be used (i.e. 8kHz), issues that makes it suitable in such applications. In addition, the algebraic subtractions between the microphone pairs offer common noise rejection and increased tolerance to omnidirectional low frequencies, reverberation noise and broadband sounds that are generated near the microphones (i.e. from the motor parts of the vehicle). Obviously, the system performs better on quitter environments that are encountered in real word rescue scenarios, while improvements are observed in audio scanning modes with the robot at standstill. On the other hand, small variations in the sensitivity, the polar and frequency response of the microphones and in the pre-amplification gains can cause errors, so that a careful calibration is needed.

Fig. 13. The prototype coincident microphone array consisting of four JOGA EM1.3 cardioid electret microphones.

The second audio module is built around the Kinect sensor, taking advantage of the integrated microphone array and A/D converters. The microphone array consists of four logarithmically spaced, cardioid microphones coupled with their corresponding A/D converters operating at 32bit /16 kHz. Hence, higher-level APIs are provided for the implementation of alternative sound source localization algorithms. In particular, cross-correlation driven time delay of arrival (TDOA) and scanning of energy-maximization through delay and sum beamforming techniques are employed to estimate the direction of incoming sound waves [22] – [23]. In this context, enhanced audio signal can be extracted for further audio event detection-validation and/or feature-based sound recognition purposes.

In addition to the above, an audio feature-based Voice Activity Detection (VAD) module has been implemented, so that when audio sound source localization is engaged, only the useful voice active segments are processed. The information regarding the estimated location of sound sources provided by the two subsystems is fused and evaluated in ROS in order to generate the final decision. In particular, four nodes are used to accomplish the audio capturing, monitoring, recording and processing procedures. The following diagram presents the adopted architecture for the audio system.

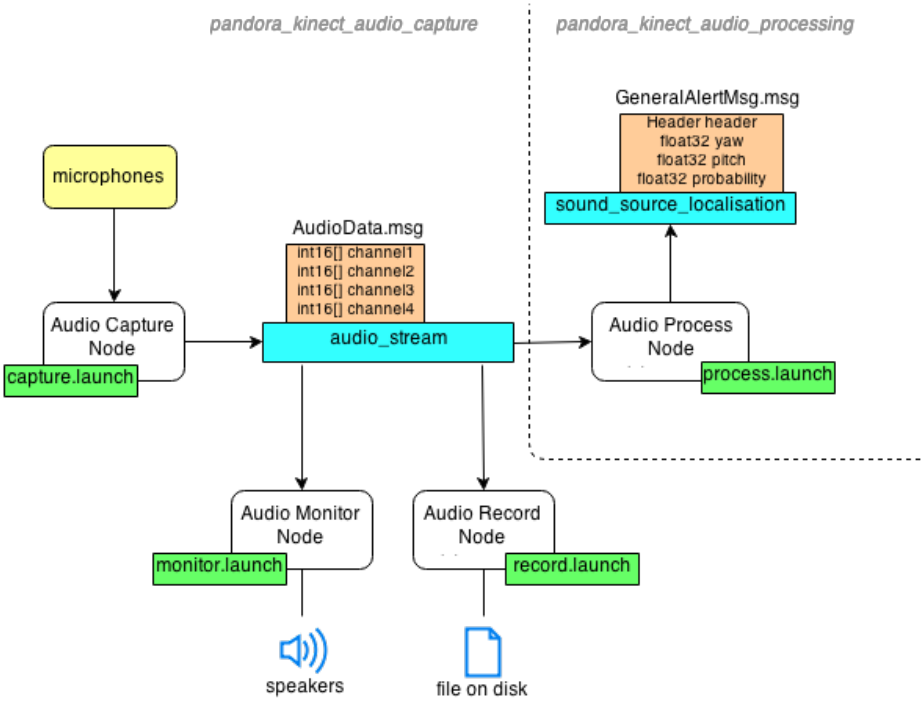

Fig. 14. Audio architecture diagram.

8. Robot Locomotion

8.1 Platform mechanical design

This year the Gears Surface Mobility Platform (SMP)[9] will be used. It is a four-wheeled all-terrain vehicle (Figure 15), with one motor per wheel. The chassis is made from aluminum. The mainboard, microcontrollers, servos, sensors and batteries are attached in a custom made chassis box using bars of an aluminum profile.

[9] http://www.gearseds.com/

The vehicle also has a sway bar, made from a titanium alloy, attached to the central chassis box that allows for the chassis box of the robot to remain in an almost horizontal plane relative to the horizon. The chassis is constructed in such a manner that the right and left wheel-set can rotate independently from the crossbar which fixates the main chassis box.

The chassis is scalable, with the ability to extend the wheelbase, but also increase the ground clearance. The weight of the robot is 8 kg. with a payload capacity of 10 kg. The size of the platform is (L*x*W*x*H) 540x590x500 mm. The platform is equipped with four 50W brushless DC motors with a reduction planetary gearhead.

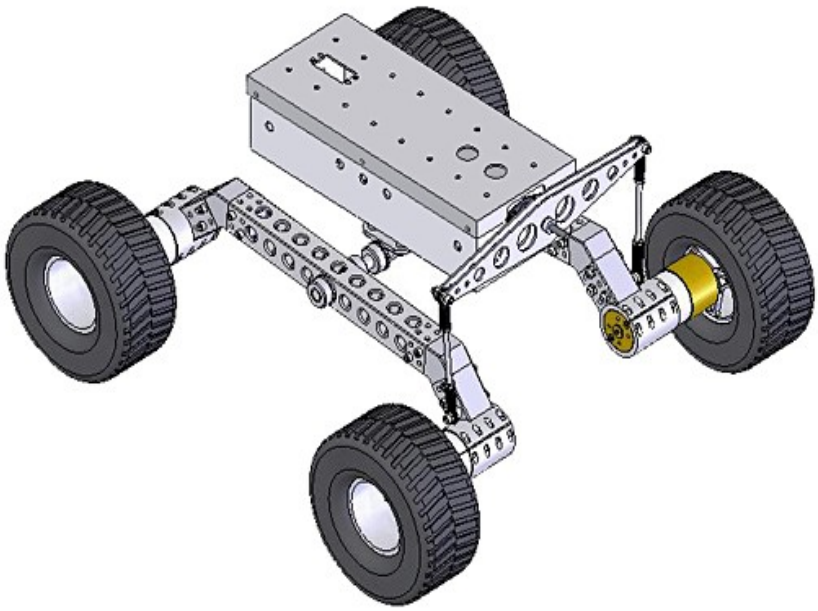

Fig. 15 The basic platform of the Gears SMP

8.2 Platform main motion control

Current architecture uses 4 MAXON Brushless DC Motors. Each motor is fitted to each wheel axle of the vehicle. The operational control of the brushless DC motors is achieved by 4 independent EPOS 2 Modules 24/3. A general review on the architecture of main motor control, including the motors is shown in Figure 16.

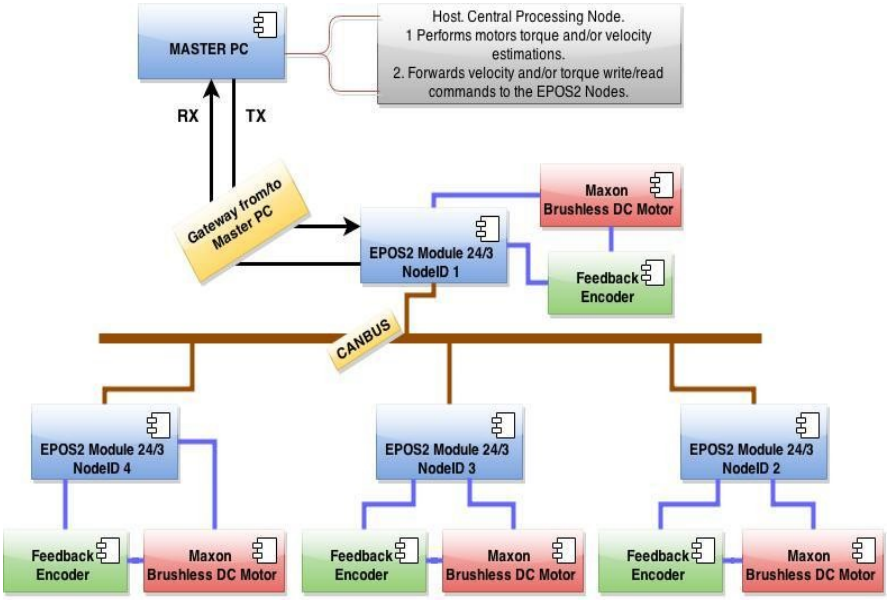

Fig. 16 Architecture of the platform main motion control

The motor controllers are connected on a CAN-BUS[10] interface and the communication is achieved by the CAN-BUS protocol. Each controller is defined as slave on the CAN-BUS interface. The communication with each slave is achieved by defining each controller with a node ID. The ID of each controller is used to send the communication frames and it is unique for every controller.
The controllers receive the commands only from one CAN-OPEN[11] Master Device that can be defined and various devices can be used to implement it. In our implementation we use as the master device the Mini-ITX. One EPOS2 controller is used to define the gateway between the master device (PC) and the CAN -OPEN slaves. In the figure the controller with node ID=1 is used to define the gateway.

The communication between the Master -PC and the node ID=1 is achieved via RS232 or USB interface. In our case the master-PC can access each one of the EPOS2 controllers and command them through the CAN-BUS communication protocol, via an RS232 interface.

[10] http://www.canbus.us/
[11] http://en.wikipedia.org/wiki/CANopen

9. Other Mechanisms

9.1 Robotic Arm

As mentioned in chapter 7, a set of heterogeneous sensors is used in order to accurately identify a victim and pinpoint his/her location. This set is mounted on a pan – tilt platform and is controlled by the Head Board. The pan- tilt platform is attached on a vertically extended arm. Thus the whole platform can be raised up to obtain a better view.

A custom linear actuator equipped with limit switches and position feedback is used. The position of the pan – tilt platform is controlled by two Dynamixel smart servos. The dc motor of the linear actuator and the servos are driven by the ARM Cortex-M4F[12] microcontroller and a Robotis CM9.04 controller respectively.

9.2 Stabilizers

Two stabilization mechanisms are mounted on the chassis box of the robotic platform. The first one allows the LRF to level, regardless of the robot's inclination while the second one controls the RGB-D sensor and the thermal camera.

The stabilization is achieved via two Dynamixel smart servos, using an Attitude and Heading Reference System (AHRS) device by PNI Sensor Corporation. It is placed on the main chassis of the robotic platform, providing heading and orientation information (Figure 17). It consists of geomagnetic sensors with gyroscopes and accelerometers. Heading measurement range is at 360 degrees, while pitch and roll at (+-) 90 and (+-)180 respectively. It provides a resolution of 0.1 degrees on all three axes. It has a heading accuracy of 0.3 degrees (rms), while pitch and roll accuracy is at 0.2 degrees (rms). It operates at 5V DC and has a continuous current draw of 60mA.

Fig. 17. The TRAX AHRS sensor

[12] http://www.arm.com/products/processors/cortex-m/cortex-m4-processor.php

9.3 Computing System (Single Board Computer)

In order to accommodate the processing needs of PANDORA, we employ a Mini-ITX system (Figure 18) placed it in the main body of the robot. The specifications of the system are the following: ibase MI980-VF mainboard, Mini-ITX, Intel QM87 Chipset, Intel i7-4700EQ Processor, 16GB of DDR3 SO-DIMMs, a Solid State Drive with 64GB capacity, all power from a M4-ATX Pico PSU. The board's dimensions are 17x17cm and for peripheral interconnection there are 8 USB ports, 5 RS-232 serial ports, a PCI FireWire, and a MiniPCIe WiFi capable add-on card with 2 pigtails for external antennas. The system power consumption is estimated at 80Watts at full computing load, without the USB, Serial and FireWire peripherals connected.

Communication between the single board computer (SBC) and the sensor network is performed through a serial interface. The higher level protocol designed allows strict timings and deterministic prediction of the CPU load generated by the sensors. Thus the robot can operate almost in real time.

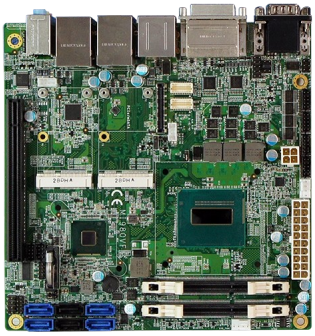

Fig. 18. The ibase MI980-VF Mini-ITX mainboard

10. Team Training for Operation (Human Factors)

The operator(s) should be familiar with the GUI and the gamepad. He/she should be able to understand the readings of all sensors and act accordingly when allowed. He/she should go through extensive training and accomplish test missions in the specially constructed arena, which emulates a destruction scene.

11. Possibility for Practical Application to Real Disaster Site

The fully deployed robotic platform has not been tested in a real environment yet. Nevertheless, the previous platform was exhibited at various technical meetings in Thessaloniki, and it was widely accepted. The Hellenic Rescue Team showed vivid interest in the potential of using the platform in real life.

In figure 19 the latest version of the PANDORA vehicle is shown and in figure 20 the vehicle's 3D model, used for simulation purposes.

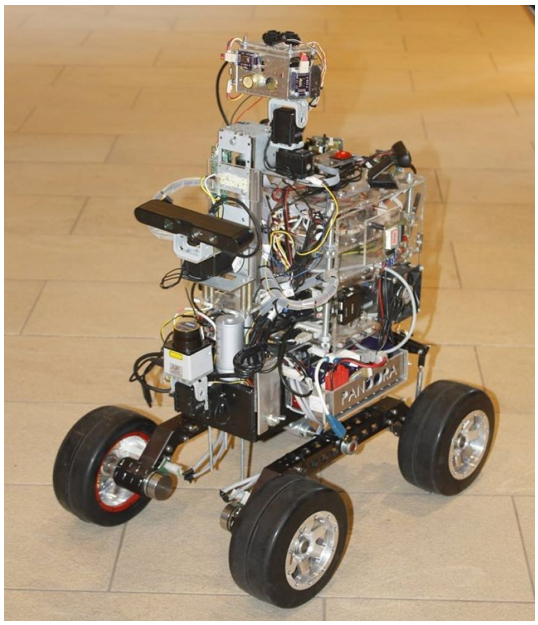

Fig. 19. PANDORA robot platform (V.2014)

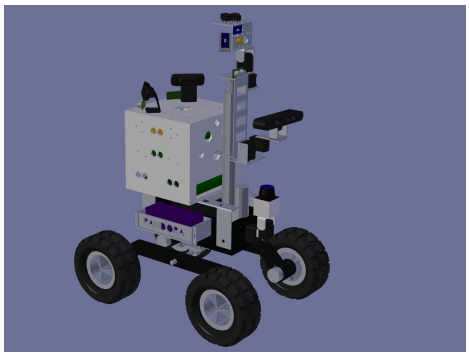

Fig. 20. 3D model of PANDORA robot platform (V.2014)

12. System Cost

The following table provides information on the cost of the parts of the PANDORA platform.

Table 2. Part names, quantities and cost

Part Name	Quantity	Price (€)	Website
Mobile Platform	1	1300	www.gearseds.com/
Chassis box	1	1000	Custom made
Arm	1	250	www.robotshop.com
Platform motors & Controllers	4	2200	www.pittman-motors.com
Servos and Controllers	6	300	www.robotis.com
LRF	1	3000	www.active-robots.com
Single Board Computer	1	1000	http://www.mini-tft.de
Sensors	20	1250	http://www.active-robots.com
Xtion Pro	1	150	https://www.asus.com/Multimedia/Xtion_PRO_LIVE/
Cameras	3	1000	www.ptgrey.com
CO_2 sensor	1	200	http://www.dynament.com
Compass	1	250	http://www.ocean-server.com
Microcontrollers & DSP	3	600	Atmel - ARM
Touch screen	1	100	http://www.olimex.com
Batteries	2	800	http://www.hoelleinshop.com
Power Supplies	2	100	
Cabling and connectors		1000	
TOTAL		**14500**	

References

1. Alonso, I.: Service Robotics within the Digital Home, Chapter on Service Robotics, Springer Verlag, 2011, ISBN 978-94-007-1491-5, pp. 89-114.
2. Quigley, M., Gerkey, B., Conley, K. , Faust, J., Foote, T., Leibs, J., Berger, E. , Wheeler, R. , Ng, A.:ROS: an open-source Robot Operating System, in ICRA Workshop on Open Source Software, 2009.
3. Emmanouil Tsardoulias, Loukas Petrou, “Critical Rays Scan Match SLAM”, Journal of Intelligent & Robotic Systems, December 2013, Volume 72, Issue 3-4, pp 441-462
4. http://wiki.ros.org/crsm_slam
5. Russell, S.J., Norvig, P.: Artificial Intelligence: A Modern Approach. 2nd edn. Upper Saddle River, New Jersey: Prentice Hall (2003)
6. Armin Hornung, Kai M. Wurm, Maren Bennewitz, Cyrill Stachniss, Wolfram Burgard, “OctoMap: an efficient probabilistic 3D mapping framework based on octrees”, Autonomous Robots, April 2013, Volume 34, Issue 3, pp 189-206
7. Sucan, Ioan Alexandru, Mark Moll, and Lydia E. Kavraki. "The open motion planning library." Robotics & Automation Magazine, IEEE 19.4 (2012): 72-82.
8. K. M. Wurm, A. Hornung, M. Bennewitz, C. Stachniss, and W. Burgard, “OctoMap: A probabilistic, flexible, and compact 3D map representation for robotic systems,” in Proc. of the ICRA 2010 Workshop on Best Practice in 3D Perception and Modeling for Mobile Manipulation, 2010, software available at http://octomap.sf.net/.
9. J. Pan, S. Chitta, and D. Manocha, “FCL: A general purpose library for collision and proximity queries,” in IEEE Intl. Conf. on Robotics and Automation, Minneapolis, Minnesota, May 2012.
10. Fox, Dieter, Wolfram Burgard, and Sebastian Thrun. "The dynamic window approach to collision avoidance." Robotics & Automation Magazine, IEEE 4.1 (1997): 23-33.
11. WillowGarage: OpenCV 2.1 C++ Reference. Available online at: http://opencv.willowgarage.com/documentation/cpp/index.html
12. Szeliski, R.: Computer Vision: Algorithms and Applications, Springer 2011 ISBN 1868-0941
13. Belhumeur, P. N., Hespanha, J., Kriegman, D.: Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection. IEEE Transactions on Pattern Analysis and Machine Intelligence 19, 7 (1997), 711–720
14. Collins R.T., Lipton A.J., Kanade T., Fujiyoshi H., Duggins D., Tsin Y., Tolliver D., Enomoto N., Hasegawa O., Burt P., Wixson L., “A system for video surveillance and monitoring: VSAM final report,

Technical Report CMURI-R-00-12, Carnegie Mellon University, 2000.
15. UgurTöreyin B., EnisÇetin A., Aksay A. and BilgayAkhan M., “Moving object detection in wavelet compressed video“, Signal Processing: Image Communication, vol. 20, no. 3, pp. 255-264, March 2005.
16. Joly, A., Buisson, O.: Logo retrieval with a contrario visual query expansion. In:MM '09: Proceedings of the seventeen ACM international conference on Multimedia, New York, NY, USA, ACM (2009) 581-584
17. Joly A., Buisson O.: A Posteriori Multi-Probe Locality Sensitive Hashing, MM’08, October 26-31, 2008. 209-218
18. Lowe, D.: Distinctive image features from scale-invariant keypoint, International Journal of Computer Vision, 60, 2 (2004), 91-110
19. Dwivedi, V. Awasthi, A Faster Non-HT Method for Detection of Circles in Digital Images for Real Time Applications, International Journal of Emerging Technology and Advanced Engineering 3, 8 (2013) 569-573.
20. Dimoulas C., Avdelidis K., Kalliris G. and Papanikolaou G., “Sound Source Localization and B-Format Enhancement Using Sound Field Microphone Sets”, Proceedings of the 122nd AES Convention, paper no. 7091, May 2007.
21. Dimoulas C., Kalliris G., Avdelidis K., Papanikolaou G., “Improved Localization of Sound Sources Using Multi-Band Processing of Ambisonic Components”, Proceedings of the 126th AES Convention, paper no. 7691, Munich, May 7- 10, 2009.
22. Thomas, Mark RP, Jens Ahrens, and Ivan Tashev.“Optimal 3D beamforming using measured microphone directivity patterns.”Acoustic Signal Enhancement; Proceedings of IWAENC 2012; International Workshop on.VDE, 2012.
23. Teachasrisaksakul, Krittameth, SurapaThiemjarus, and ChantriPolprasert.“A Bayesian Approach for Sound Source Estimation.”Ecti Transactions On Computer And Information Technology Vol.7, No.2 November 2013.