RoboCup Rescue 2007 - Robot League Team <MRL (IRAN)>

F. Barazandeh[1], M. Norouzi[2], S. M. Mavaei, A. H. Mashat , J. Chegini, A. Byagowi,
M. Yaghobi, A. Paikan, M. Jadalihi, B. Karasfi ,J. Zolghadr, M. Moshfeghi

[1] fbarazandeh@aut.ac.ir [2] mh_norouzi@ee.iust.ac.ir

Mechatronics Research Laboratory
Azad University of Qazvin, IRAN

Abstract. In this paper the MRL rescue robot team and its robots are explained. We have designed and built two new robots including one autonomous indoor robots and one out-door robot for different situations/arenas. Our main goal of this activity is to achieve a practical rescue robot for real situation such as earthquake which is quite common in our country. We have also arranged to initiate some research programs on autonomous mobile robot such as; simultaneous localization and mapping, navigation strategies, exploration and motion planning, sensor fusions, visual odometry and search algorithms. All of our research works are carried out at the Mechatronics Research Laboratory (MRL).

Introduction

Rescue operation in a disaster situation is quite important and should be fast enough to save victims life. So implementing high technologies such as robotics could be quite helpful for search and rescue operations. There are so many earthquakes every year in many countries such as Japan, USA, Turkey and Iran. Robocup real rescue competition has provided quite stimulating situation for University educator to involve in such a humanitarian activity.

Our team members in Mechatronics Research Laboratory are planning not only to take part in the competition but also to get enough knowledge to achieve a practical robot to help search and rescue operations in a disaster situation.

In this paper the MRL rescue robot team and its robots are explained. We have designed and built two new robots including one autonomous indoor robots and one out-door (rough terrain) robot for different situations/arenas. Our main goal of this activity is to achieve a practical rescue robot for real situation such as earthquake which is quite common in our country. We have also arranged to initiate several research programs on autonomous mobile robot such as; simultaneous localization and mapping, navigation strategies, exploration and motion planning, sensor fusions, robot cooperation, visual odometry and search algorithms. All of our research works are carried out at the Mechatronics Research Laboratory.

It should be mentioned that our previous robots such as NAJI-II which took 1st place in German open 2005 and Iran Open 2006 are still active and running.

Also NAJI-V which took 2nd in Bremen 2006 in Locomotion Challenge are still active and running.

Obviously, based on the environmental situation a special robot with proper abilities is required. In other words, there could be no unique robotics solution for searching and rescuing program in a disaster situation. As a result we have designed different robots with different maneuverability. For example NAJI-II and NAJI-V with a high power and flexible mechanism which overcome hard obstacles are also capable of supporting a powerful manipulator for handling objects. Fig.1 illustrates NAJI-II in red arena in German open-2005.

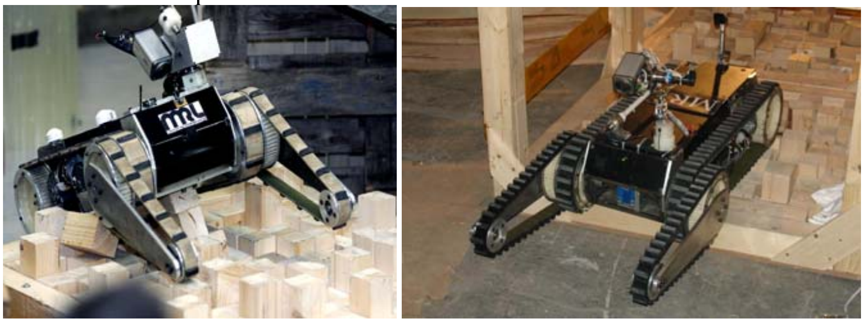


Fig 1 NAJI-II Climbing Step Field in German Open-2005 and Iran Open-2006

NAJI-V was a new design and modified version of NAJI-II which is more powerful and flexible while it is lighter and smaller. A 3D Model of NAJI-V is Illustrated Fig, 3. Our new design was inspired from our acquired experience on NAJI-II and NAJI-V in German open-2005, Robucup2005 in Osaka, Iran Open 2006 and Robocup 2006 in Bremen. There are so many rough and hard terrains in a disaster situation which the rescue robot should be fast enough and low weigh to pass and explore environment quickly while it is stable.

Fig 2. Illustrated the new stylish of NAJI-II in Iran Open 2006 which took 1st place.

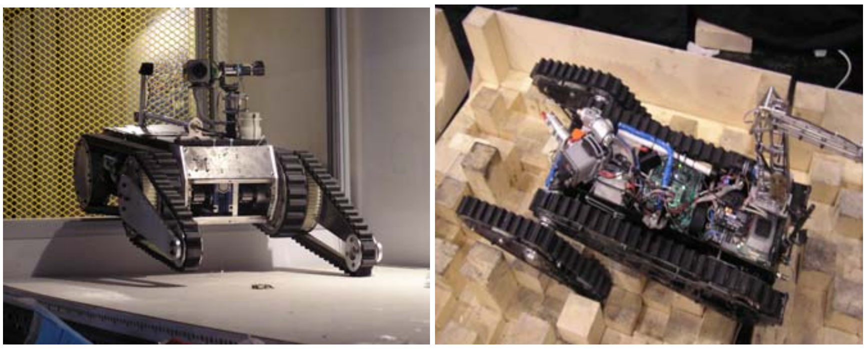


Fig 2 NAJI-II with new stylish and NAJI-V in Robocup 2006

NAJI-V with the new stylish is now very stable and more efficient than later.

And using new Mechanical stylish in NAJI-V makes this robot more power full and effective in Step-Filed zones as later.

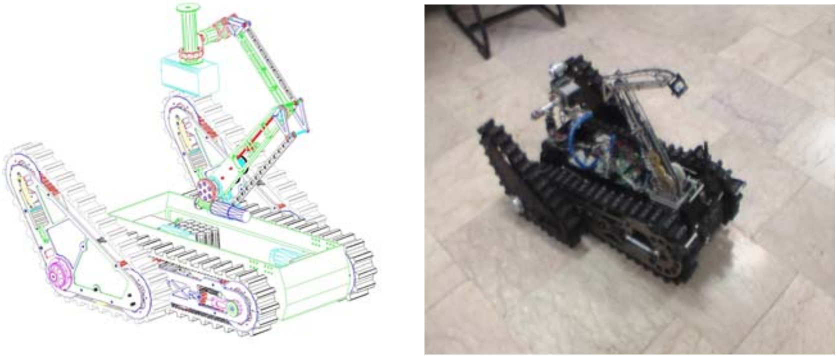


Fig 3 3D Model and Real Picture of NAJI-V

NAJI-II and NAJI-V are good examples of such a robot while NAJI-V with a novel mechanical design is quite faster, flexible and more stable. NAJI-IV which is facilitated with most required sensors is an autonomous mobile robot to carry out different research programs which is also suitable for the yellow arena.
Each robot is powered by Embedded PC based on RT-Linux for real time data processing and robot controlling .

In fact NAJI-VII is a combination of NAJI-II and NAJI-V. This new design has the ability of NAJI-II for climbing and the ability of NAJI-V to move over Step-Fields, and other special specification. Fig 4 shows 3D model of NAJI-VII and in Fig 5 you can see some of abilities of this new robot. We are developing a new autonomous robot that can be seen in Fig 6.

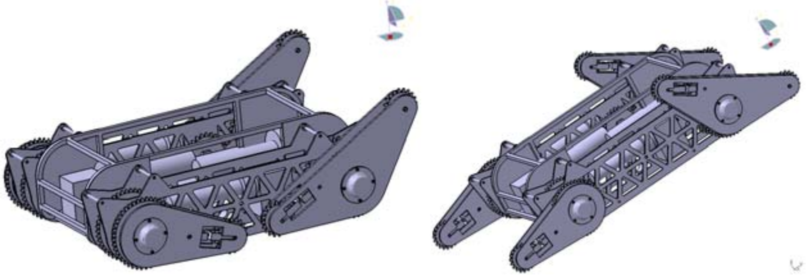
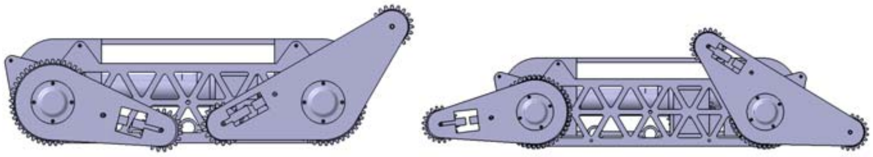


Fig 4 3D Models of NAJI-VII new Robot



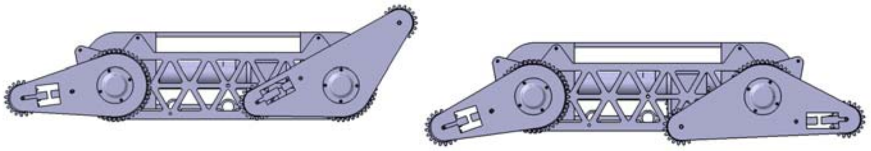


Fig 5 some abilities of NAJI-VII new Robot

And in Fig 6 you can see the latest ability of NAJI-VII that can moves the robot like a Four Legged robot.

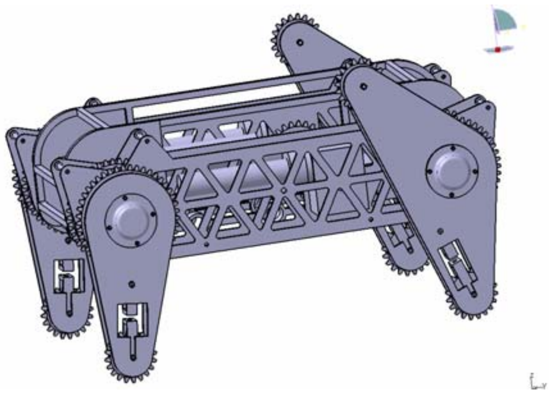


Fig 6. Latest abilities of NAJI-VII new Robot

1. Team Members and Their Contributions

Organization:

F. Barazandeh	Team leader
M. Norouzi	Team organizer

Electronic and Control:

S. M. Mavaei	Electronics design / Instrumentation
A. H. Mashat	Electronics design / Instrumentation
J. Chegini	Electronics design / Sound Processing
A. Byagowi	Electronics design / Hardware Vision

Software and Algorithm:

M. Yaghobi	Software developer
A. Paikan	Operator / Software developer
M. Jadalihi	Localization Algorithms
B. Karasfi	Exploration Algorithm

Mechanical:

J. Zolghadr	Instrumentation / Mechanical design
M.Moshfeghi	Mechanical design
M.Rahmani	Mechanical Technician
M.Ghaffari	Mechanical Simulation

Advisor:

- Dr. M. Moosakhani
- Dr. A. T. Haghighat

Sponsor:

Azad University of Qazvin	Sponsor

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

As it is compulsory to set-up and break down in less than 10 minutes, we are going to consider several things for fast handling and operation. We are going to design a Mobile Control Pack (MPC) including; notebook, joystick, access point, antenna, I/O Extension board and case with appropriate connectors. .

Our operators are practicing more and more in a real situation to do everything as fast as possible. A good joystick with several control keys and handles is selected for easier and faster operation. It should be noted that we are educating two operators to replace any of them in case of happening any problem. you can see our operator driving robot in Fig 7

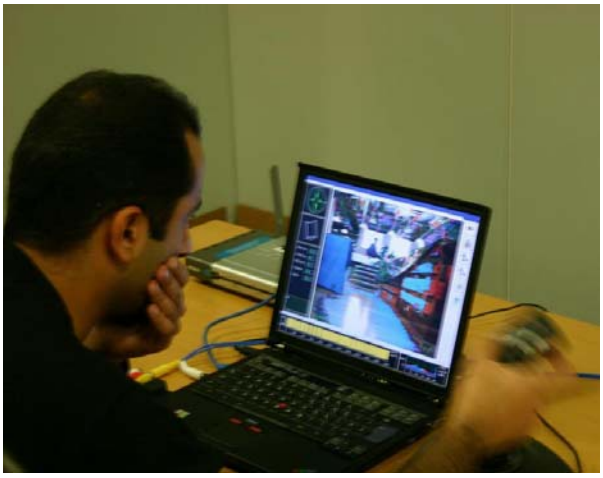


Fig. 7 Control Pack and Operator that's running the Robot

Our operators are equipped with a communication headset to contact their team members in case of requiring anything or organizing the set-up and break down process just on time. (This headset is just for the set-up and break-down tasks).

3. Communications

As it is suggested, we are going to use Wireless LAN IEEE 802.11a (5GHz).

Our communication system is based on the higher frequencies and Channels of IEEE 802.11a Standard. Our communication data consist of Robot commands, Digital audio/ video and sensors data. Two Access Point/Bridge with two external antennas on Channel 161 (frequency 5.805 GHZ) are applied as our communication system.

Rescue Robot League		
MRL (IRAN)		
Frequency	**Channel/Band**	**Power (mW)**
5.805 GHz - 802.11a	161 / A	100

4. Control Method and Human-Robot Interface

According to previous real rescue leagues, it seems that there is a trend to make the robot as intelligent and autonomous as possible. Now it is a rule that all robots in yellow arena should be fully autonomous. Accordingly, it is our plan to manage and prepare for an intelligent robot capable of navigating, mapping and searching for victims automatically. Therefore, we have generally designed two types of robots; semi-autonomous (NAJI-II & NAJI-V & NAJI-VII) and fully autonomous (NAJI-IV) which are presented in the following sections. Concerning system overview both types are mostly similar. In other words the difference between two types is the inference of human operator in semi-autonomous robot to command and navigate the robot and search for victims. Even our intelligent collision avoidance algorithm has more priority than the command of the operator.

4.1 Software Overview

Our software controller is developed to be executed on Real-time Linux. Moreover, it is equipped with intermediate software layer to communicate with RT-HAL in lower level and Wireless LAN in higher level. The robot sends the generated path, Log file and sensors data to central computer in the operator station. Goals of our software are to develop source code capable of executing in the optimized real time Linux and communicate with RT-HAL and Wireless LAN.

4.1.1 Real Time Hardware Abstracted Layer (RT-HAL)

It should be noted that for executing a high level control algorithm in a robot and consequently decreasing the system dependency on intermediate devices, employing an abstracts layer is unavoidable. To approach this goal the Real Time Hardware Abstracted Layer (RT-HAL) is designed in modular form on Linux Kernel. RT-HAL in order to directly access to the lower layer (hardware) and upper layer (high level controlling process). This process can be broken down into a number of important sub tasks which are:

- Collecting the sensors data
- Sensor fusion and data filtering
- Low Level controlling such as motion planning
- Self localization and exploration
- A communication interface to high level process

Sensors data will be collected by associated data acquisition module and passed to sensor fusion module to be fused and acquire a better perception of environment. This data will be analyzed by Self Localization and Obstacle Detection (SL/OD) to obtain the robot and obstacles pose. Sensors data will be sent to sensor fusion module for error correction and sensor fusion algorithm. Output data such as robot rotation

(▯▯), movement (▯d) and obstacles distance will be sent to SL/OD and motion planning module to further process. For example, the encoder's pulses which are used to determine the angle of rotation will be fused with IMU data, and consequently more accurate angle of rotation will be achieved.

Moreover, this module has an interface which uses the FIFO to transmit these data to higher layer. SL/OD module periodically, places data into the FIFO and the high level application obtain its required data from FIFO. RT-HAL by implementing the driver as a device files, prepares a base for high level application to send their controlling command to actuators. Consequently RT-HAL decreases the high level application independency and prepares the standard API to communicate with the hardware which leads to code the application in any programming languages. Fig 8 illustrates the RT-HAL layers.

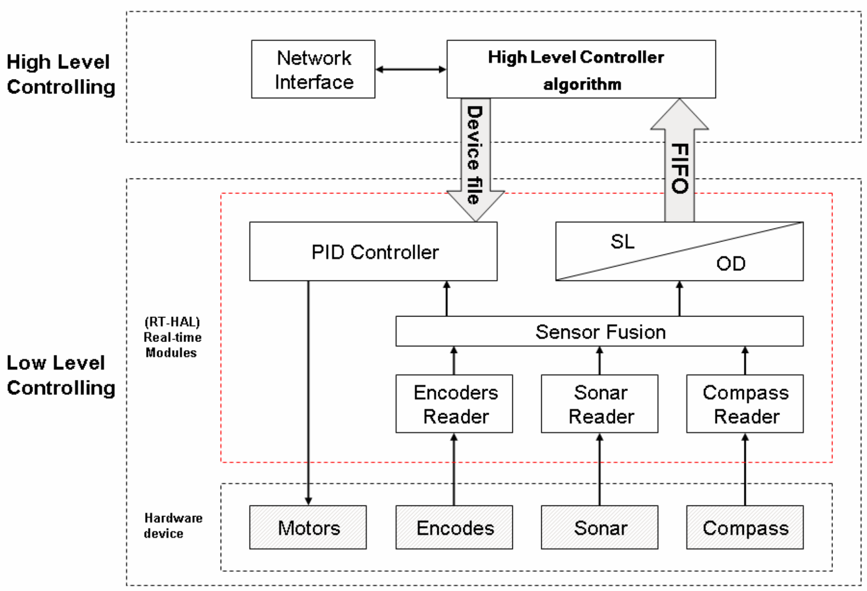


Fig. 8 RT-HAL layer overview

4.2 Hardware Overview

Our robots which are based on differential drive system are equipped with sonar sensor, Laser Scanner, CO_2 sensor, thermopile array detecting infra-red sensor, IMU, digital compass, optical encoders and digital/analog cameras. The computational system is PCM 6892 and PCM 8200, 512 MB Compact Flash Memory, 128 MB RAM, IO Card with 32 digital IO ports and Wireless LAN 802.11a/g you can see a hardware overview in Fig 9.

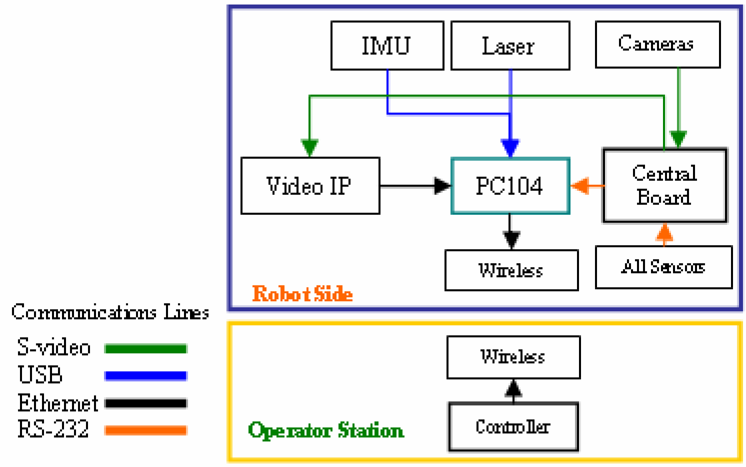


Fig. 9 Hardware Overview

4.2.1. Robot System Overview

Our robot system overview is designed based on three levels High, Abstraction and Low level top to bottom including the following steps orderly;

- Localization: Given sensors and a map, where am I?
- Vision: If my sensors are eyes, what do I do?
- Mapping: Given sensors, how do I create a useful map?
- Searching Algorithms: Given an unknowable world but a known goal and local sensing, how can I get there from here?
- Kinematics: if I move this motor somehow, what happens in other coordinate systems?
- Motor Modeling: what voltage should I set now?
- Control (PID): what voltage should I set over time?

The control scheme of our autonomous robot is partially autonomous. It means that the cameras images are sent to the computer are processed by operator to navigate the robot. All other sensors information are also sent to the operator to investigate the arena and detect all possible victims. To avoid colliding the robot with the environment obstacles or victims an ultrasonic obstacle avoidance algorithm has more priority than the command of the operator.

Although the map generation is autonomous, when a victim is located, operator has to define the victim conditions based on the sensors data. In order to save time, a proper GUI is designed with several push bottom keys to define the victim's condition just by clicking the mouse button.

In case of loosing the control of the robot by the operator, a program is designed to return the robot back to the starting point autonomously using the map stored in the robot. All the sensors data are collected in a data bank to be used even off-line after the operation.

It should be mentioned that the applied servomechanism is digitally controlled by a PID control algorithm implemented on an AVR microcontroller system. An optical shaft-encoder is used to feedback the position and speed of the robot. All embedded microcontroller system designs for servomechanisms, instrumentations and drivers are modular for easy repair and maintenance.

5. Map Generation/Printing

Map generation method in our semi-autonomous robot (NAJI-II & NAJI-V) is based on the operator assessment in conjunction with the collected data and a GUI program, which enables operator to locate and register different object such as victims, obstacles, walls and doors. We use LMS400-1000 in autonomous and URG-X003 in semi-autonomous robot. However, NAJI-IV generates 2D and 3D maps automatically. As the robot proceeds to search and identify the field, the map of the environment will be developed which might be printed at the end of the robot mission. In our program, two maps will be generated. First, a 3D-map of the robot path with the location of all the victims will be generated autonomously. In this method, environmental conditions will be recorded by LMS400-1000 laser scanner and a set of 12 SRF08 sonar sensors which are distributed around the robot. It should be noted that each of these two types of sensors are necessary to fulfill the deficiency of the others.

In NAJI-II, while the robot is navigating the arena, the victims' condition will be recorded by the operator decision in the victims' data bank. At the end of the operation the victims' conditions will be printed by selecting each victim individually as the second map. Fig 10 illustrates a sample page of the NAJI Rescue Robot camera.

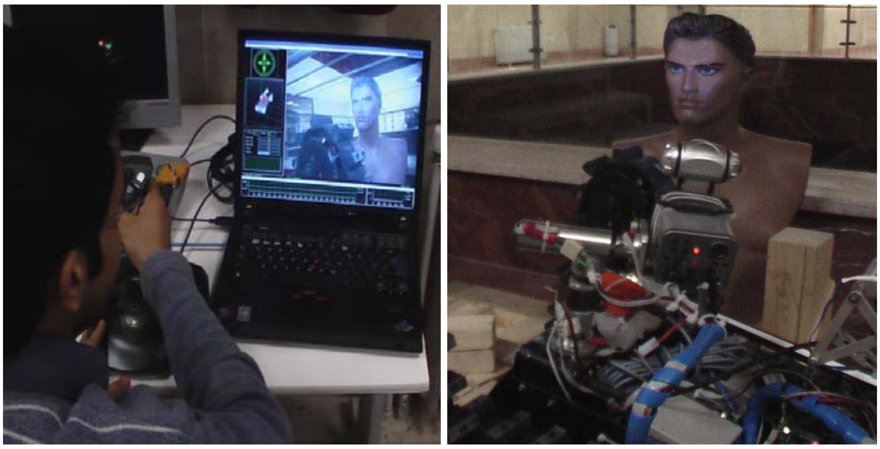


Fig. 10. A sample page of NAJI Rescue Robot GUI

5.1. 2D/3D Map Generation Using Laser Scanner

Our goal is to generate a 2D/3D map from the environment which the autonomous mobile robot has navigated. A laser scanner from HOKUYO Automatic Co. [12] is used. URG-X003 is a laser sensor for area scanning. The sensor's light source is infrared laser of wavelength 785nm. Scan area is 240° semicircle with pitch angle 0.36° which is good enough for indoor applications. Sensor outputs the distance measured at every point. The sensor's maximum measurement is up to 4000 mm with the accuracy of 1 mm.

At 4000mm, the laser beam diameter is 40mm. The principle of the distance measurement is based on calculation of the phase difference, due to which it is possible to obtain stable measurement with minimum influence from object's color and surface gloss. URG-X003 is designed under JISC8201-5-2 and IEC60947-5-2 standards for the industrial applications such as obstacle detection. Fig. 8 illustrates the coverage area by URG-X003.

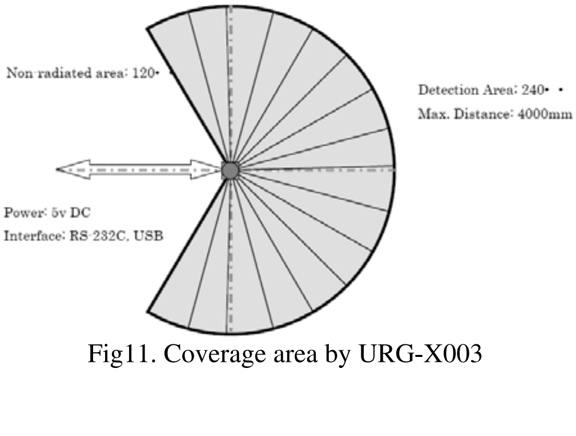


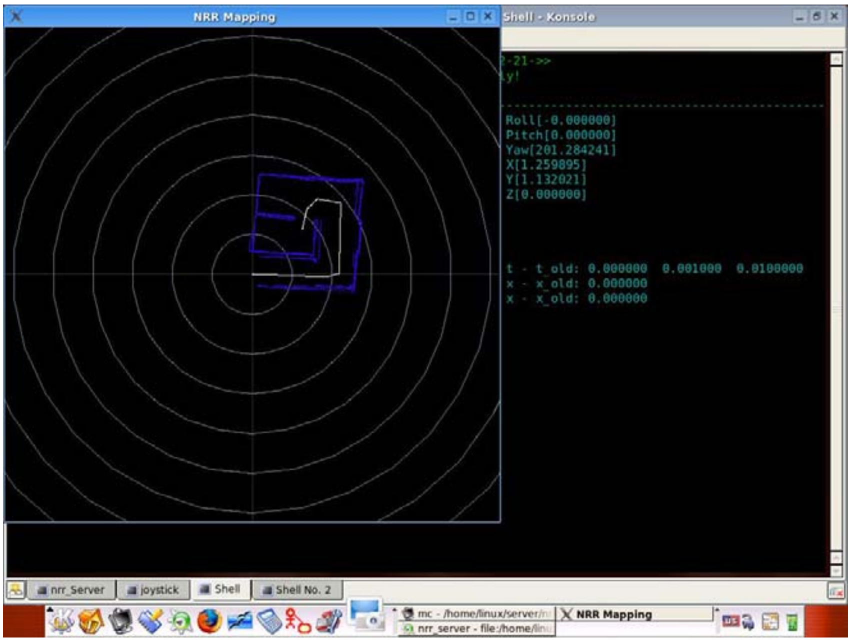
Fig11. Coverage area by URG-X003

The laser scanner system is located in front of the robot which is able to scan 240° semicircle ahead of the robot. In order to filter the thermal noise of the laser, a simple statistical method based on variance is applied. The odometry, digital compass and laser scanner data are fused to determine the location and heading angle of the robot with 0.5 degree resolution. According to this data the environment map will be generated.

By sampling the laser data and applying it to the rotation matrix which varies according to the angle of the robot heading, it is possible to have an absolute scan which is not dependant to the robot heading angle. It means that if the robot rotates around its center of gravity a few rounds while scanning the environment, the data calculated from rotation matrix for each scan are mostly equivalent. So this is the first and important step to generate 2D map.

Now, it is possible to merge the scanned data using different methods such as SLAM [7], Bayesian and Kalman filter to have 2D map. Using a tilt sensor (ADXL202) it is possible to distinguish between motion of the robot on a slope or a flat area which is important not to make mistake in generating 2D map.

In order to generate a 3D map we have used two different structures for NAJI-II and NAJI-V. First we have designed a servomechanism to rotate the laser scanner vertically. For our fully autonomous robot (NAJI-IV), we have attached the laser scanner to the back of the robot while the laser beam is toward the ground. By this way, when the robot is moving forward and laser scanner is scanning at the same time, we may have a 3D map.



```
2-21->>
y!
-----------------------------------------------------------
Roll[-0.000000]
Pitch[0.000000]
Yaw[201.284241]
X[1.259895]
Y[1.132021]
Z[0.000000]



t - t_old: 0.000000   0.001000   0.0100000
x - x_old: 0.000000
x - x_old: 0.000000
```

Fig12. Generated 2D Map on NAJI-IV

We have used a simple calculation of the neighbor points gradient as a segmentation method to distinguish between walls, ground surface and objects. Fig. 9 illustrates a sample of generated map.

6. Sensors for Navigation and Localization

In order to navigate the semi-autonomous robot in an unknown environment, a set of two cameras are used to help the operator to command/guide the robot and also a set of ultrasonic sensors to detect unknown obstacles which can be used in an automatic collision avoidance algorithm. Also we have used IMU (Inertial Measurement Unit) and optical encoder for localization. IMU determines Heading (Roll & Pitch & Yaw) of robot in addition Fuse between IMU with odometry leads to decrease of incremental errors.

Our Autonomous robots are equipped with the following sensors for localization and navigation;

- Optical Shaft Encoder,
- Ultrasonic Sensors (SRF08) with range of 6 meters to locate obstacles, [13]
- Laser Scanner, [12]
- Digital Compass to measure the robot heading angle, [13]
- Inertial Measurement Unit [15] to fuse its data to reduce odometry error and Heading
- A set of Ultrasonic sensors for co-operation of two robots in order to search and navigate an environment with minimum error. [13]

6.1. SLAM based on Fusion of Sonar and Laser Scanner

Autonomous mobile robots require two types of algorithms for exploration and mapping, in an unknown environment; [7]

- Localization and mapping algorithm that extracts features of object from row sensors data to map the environment and identify pose of robot in that map.
- Exploration algorithms which guide robots to uncovered parts of map. [8]

.

6.1.1 Localization and Mapping [5]

Localization and mapping in cluttered and unstructured environment require fusion of row sensors data to reduce the uncertainty of each sensor in order to fulfill individual sensors properties. Several approaches are applied in sensor fusion, including conventional methods based on statistics theorem such as; Bayesian estimation, dempster-shafer inference, linear and extended Kalman filter, as well as intelligent algorithms such as; fuzzy logic and genetics algorithms.

Moreover, localization and mapping may need to be performed in two different methods;

1. Concurrent Localization and Mapping

Firstly, robot maps the existing around objects, and then moves to reach its intermediate targets (set point). Consequently, it estimates its pose and modifies the generated map iteratively.

2. Simultaneous Localization And Mapping (SLAM) [7]

   SLAM is a technique used by robots to build up a map within an unknown environment while at the same time keeping track of its current position. This is not as straight forward as it might sound due to inherent uncertainties in discerning the robots relative movement from its various sensors. In this method, map is presented by a set of geometrical objects while pose estimation of the robot and objects are updated in a state space matrix iteratively. SLAM realization could be made by Monte-Carlo, Markov or Extended Kalman Filter (EKF). [8]

   As we have implemented our SLAM algorithm by EKF, this method is described below (Fig. 13.);

   In this method, each estimation has an associated covariance matrix which expresses the correlation between variables in state space matrix. This covariance matrix also describes the estimation uncertainty. So we have to calculate covariance matrix in each iteration.

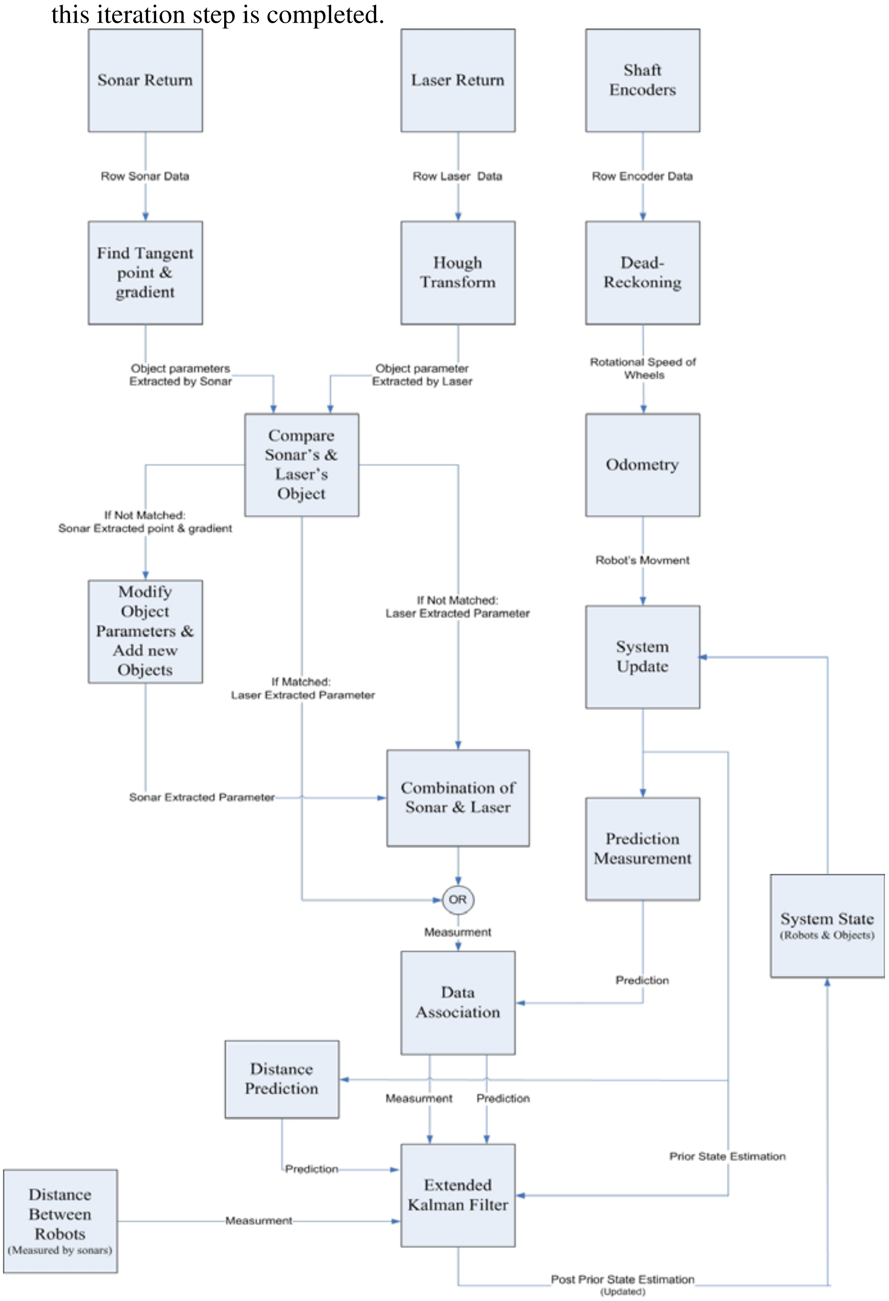
   Firstly, dead reckoning method based on shaft encoder data, calculates rotational speed of robot's wheel. In observation step, the robot acquires sonar's and laser scanner dataset, then environmental objects geometrical parameters could be determined by separate feature extraction algorithms for sonar and laser scanner data. Laser scanner feature extraction could be performed by Hough transform [9] to fit lines to the laser scanner return points. Because of wide angular range, feature extraction for sonar's data is more complicated. The method that we have used in our robot is the same as Leonard's research work which is described in "Direct Sonar Sensing". [6]

   Using two sensor types for mapping in an unknown environment has an essential benefit, which returns in physical principals of sensors. For example laser scanner couldn't detect dark object perfectly, and mostly it returns distance of those object with an offset. In contrast, sonar could detect dark object as well as bright ones, but sonar has a wide angular range that limits detection angular resolution. Beside ultrasonic sonar sensing is slower than laser and other optical sensing system. With respect to these reasons, we have used sonar and laser scanner which have their own individual feature extraction procedures. After calculating objects parameters, these parameters must be compared to find out which parameters belong to the same object and which ones are different. If sonar and laser data detect same object with minor differences in parameters, mostly, laser's data are more suitable than sonar's. Generally it could be presumed that laser's data are more precise than sonar's.

   But while differences between extracted parameters are bigger than a threshold, we assume that these parameters belong to different objects and should run data association procedure.

   Data association is a matching procedure between sensors measurement and predictions from previous estimation of object locations existing in map, to assign one prediction to each sensor measurement. If a sensor measurement does not match to any predictions, it means that this object is observed for the first time as a new object, and it should be added to the previous existing objects in map. Realization of data association in our robot is performed by Nearest Neighbor Standard Filter (NNSF). [10]

Based on innovations and measurement covariance matrixes, Kalman filter gain could be find simply and finally, the post prior state estimation is updated and this iteration step is completed.

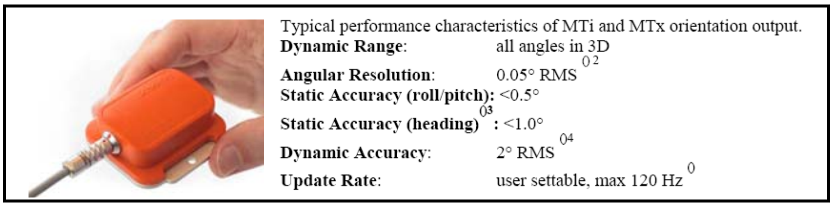


Sonar Return
Laser Return
Shaft Encoders
Row Sonar Data
Row Laser Data
Row Encoder Data
Find Tangent point & gradient
Hough Transform
Dead-Reckoning
Object parameters Extracted by Sonar
Object parameter Extracted by Laser
Rotational Speed of Wheels
Compare Sonar's & Laser's Object
Odometry
If Not Matched: Sonar Extracted point & gradient
Robot's Movment
Modify Object Parameters & Add new Objects
If Not Matched: Laser Extracted Parameter
System Update
If Matched: Laser Extracted Parameter
Sonar Extracted Parameter
Combination of Sonar & Laser
Prediction Measurement
System State (Robots & Objects)
OR
Measurment
Prediction
Data Association
Distance Prediction
Measurment
Prediction
Prior State Estimation
Prediction
Extended Kalman Filter
Distance Between Robots (Measured by sonars)
Measurment
Post Prior State Estimation (Updated)

Fig.13. SLAM Algorithm Flowchart

6.2. Fuse of Inertial Navigation System and Odometery

In our fully autonomous robot we are working on a research project to fuse the odometery data and an Inertial Measurement Unit data to reduce summing odometery errors. The applied IMU is from Xsense Company which is shown in Fig. 14.



Typical performance characteristics of MTi and MTx orientation output.

Dynamic Range:	all angles in 3D
Angular Resolution:	0.05° RMS [02]
Static Accuracy (roll/pitch):	<0.5°
Static Accuracy (heading) [03] **:**	<1.0°
Dynamic Accuracy:	2° RMS [04]
Update Rate:	user settable, max 120 Hz [0]

Fig.14MTi Sensor from Xsense Company

As the robot velocity is quite slow and sometimes it is completely stationary, we have experienced a few problems regarding the biased data of accelerometers inside the IMU. Besides we have to integrate twice from the accelerometer to find the position of the robot, bias voltage would be so disturbing. To solve this problem we have proposed a fuzzy algorithm to reset the summing errors related to the biased output of the IMU [3]. Fig. 15 shows the algorithm.

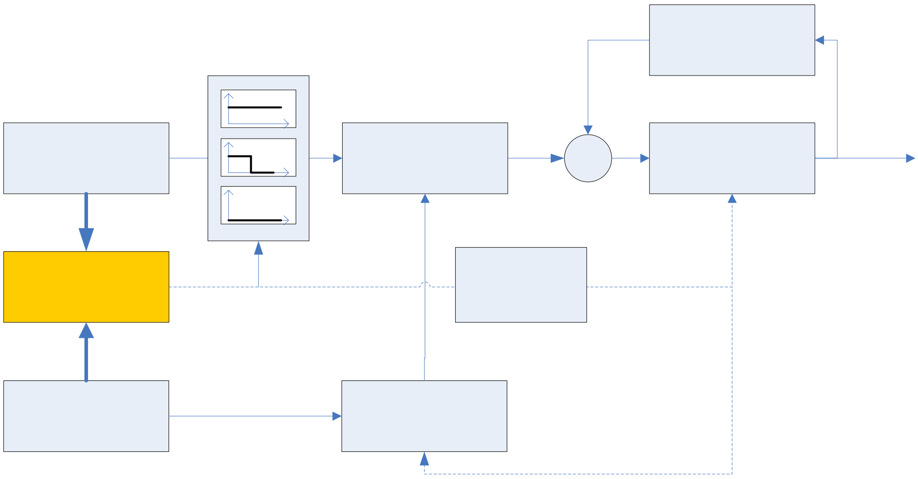


Fig15. Fuzzy Algorithm to Fuse IMU and Odometery Data

The main idea of this navigation algorithm is based on use of “tightly coupled” configuration of odometry and INS data by a fuzzy decision maker algorithm [1], [2], [4].

7. Sensors for Victim Identification

For victim identification we have used a set of different sensors which are explained below;

- TPA81Thermopile Array to measure Victim's temperature [13]

The TPA81 which is shown in Fig. 16 is a thermopile array detecting infra-red in the 2um-22um range. This is the wavelength of radiant heat. These Pyro-electric sensors can only detect a change in heat levels though - hence they are movement detectors. Although useful in robotics, their applications are limited as they are unable to detect and measure the temperature of a static heat source. The TPA81 can also control a servo to pan the module and build up a thermal image. The TPA81 can detect a candle flame at a range 2 meters (6ft) and is unaffected by ambient light.

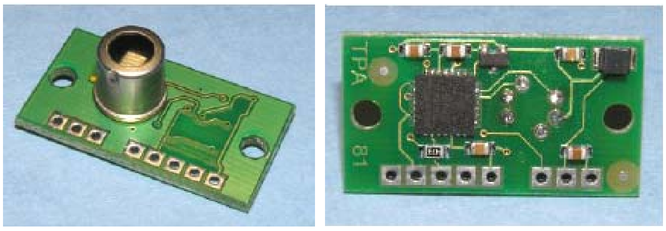
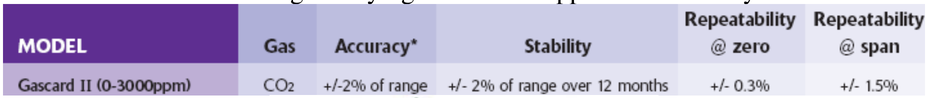


Fig16 TPA81 Thermopile Array

- Infrared Gas Sensors (Carbon Dioxide CO_2 Sensor) [11]

The sensor that we have used to sense the exhale CO_2 from victims is Cascard II module (Fig. 17) from Edinburgh Instrument Co. The Gascard range of OEM infrared gas sensors offer excellent quality detection and measurement of many gases, using pumped or diffusion aspiration technology to provide rapid instrument response characteristics. The Gascard II range provides high accuracy detection and measurement of CO2, CH4 (methane), C6H14 (hexane), and other hydrocarbon gases, with available measurement ranges varying from 0-2000ppm to 0-100% by volume.



MODEL	Gas	Accuracy*	Stability	Repeatability @ zero	Repeatability @ span
Gascard II (0-3000ppm)	CO_2	+/-2% of range	+/- 2% of range over 12 months	+/- 0.3%	+/- 1.5%

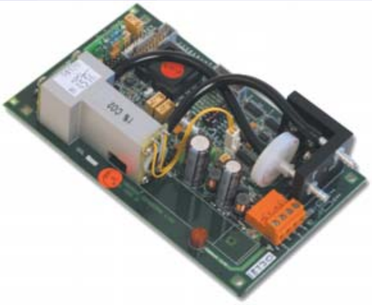


Fig17 Carbon Dioxide Module

We have also used a set of Cameras to identify the victim's tag, motion and its gesture. A Microphone is also used to detect victim's sound.

8. Face Detection

Face detection is an important element of various computer vision areas, such as image retrieval, shot detection, video surveillance, autonomous Rescue Robot and etc.

We are using a face detector algorithm that based on Haar-like features; our implementations had a high degree of false positives, and detected the same face multiple times.

The goal is to find an object of a pre-defined class in a static image or video frame. Sometimes this task can be accomplished by extracting certain image features, such as edges, color regions, textures, contours, etc. and then using some heuristics to find configurations and/or combinations of those features specific to the object of interest. But for complex objects, such as human faces, it is hard to find features and heuristics that will handle the huge variety of instances of the object class (e.g., faces may be slightly rotated in all three directions

an individual location within the image by using the classifier cascade to find whether it contains a face or not. Helper functions calculate integral images and scale the cascade to a different face size (by scaling the coordinates of all rectangles of Haar-like features) etc.

A recognition process can be much more efficient if it is based on the detection of features that encode some information about the class to be detected. This is the case of Haar-like features that encode the existence of oriented contrasts between regions in the image. A set of these features can be used to encode the contrasts exhibited by a human face and their special relationships. Haar-like features are so called because they are computed similar to the coefficients in Haar wavelet transforms .Fig. 18 illustrates a library of faces that used by our algorithm. You can see a real sample of face detection in Fig19.

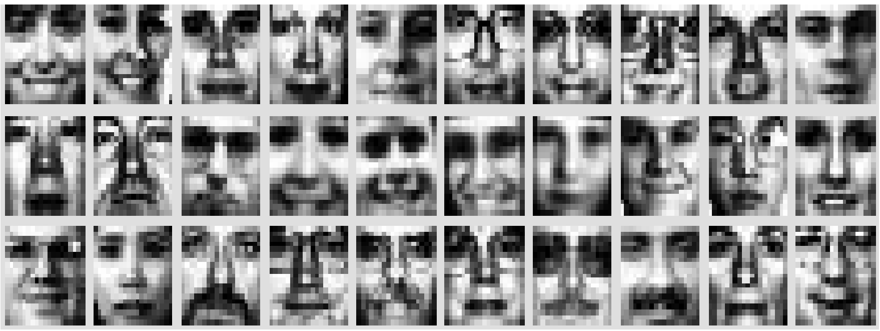
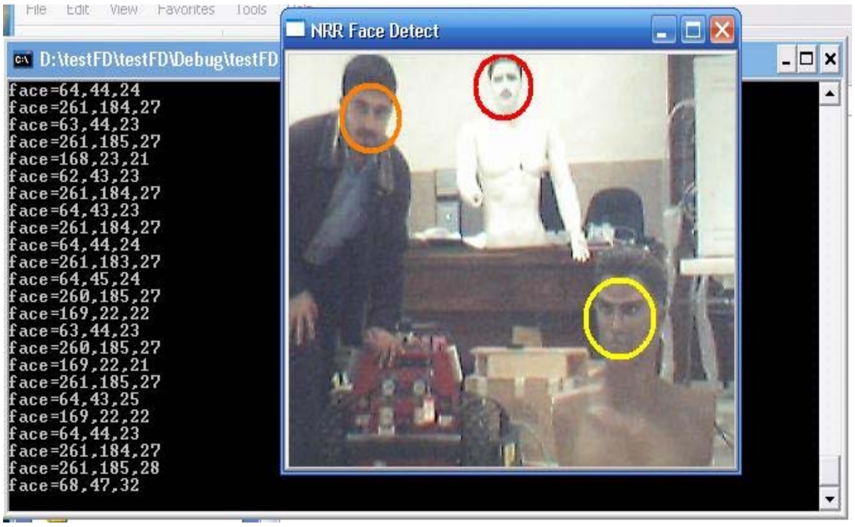


Fig18.(a library of faces)



D:\testFD\testFD\Debug\testFD

```
face=64,44,24
face=261,184,27
face=63,44,23
face=261,185,27
face=168,23,21
face=62,43,23
face=261,184,27
face=64,43,23
face=261,184,27
face=64,44,24
face=261,183,27
face=64,45,24
face=260,185,27
face=169,22,22
face=63,44,23
face=260,185,27
face=169,22,21
face=261,185,27
face=64,43,25
face=169,22,22
face=64,44,23
face=261,184,27
face=261,185,28
face=68,47,32
```

Fig19.result of face detection algorithm

10. Visual Odometry

Visual odometry is the process of determining the position and orientation of a camera by analyzing a sequence of images from it. This can be useful for vehicle and robot navigation in locations where other techniques can not be used.

This method suffers from many problems, such as:

- The precision depends on very exact measurements of the robot's wheels and wheelbase
- The surface the robot runs on must be smooth
- The wheels can slide on the floor
- The error in the position is accumulating

The input to a visual odometry system comes from one or more digital cameras. There are

many different kinds of camera configurations. Things that may differ are:

- Resolution
- Auto focus
- Zoom
- Lens
- Sensor type

The precision of the visual odometry system depends, among other things, on the resolution of the camera images. If the resolution is high the precision can become better, but the system has to handle more information. More information can lead to more errors in the object matching process, because of problems with more noise in the image. The refresh rate of the system can be reduced, due to longer processing time per frame.

When the object points are reconstructed, the system needs to make a temporary match of the points. From the matched points rigid body transformation of the two point clouds can be computed. The parameters from the rigid body transformation are used to calculate the camera motion despite the limitations in the hardware the system performed very well. The environment in which the system works is limited due to problems in the spatial matching. The system produces a lot of information about the 3d position of object points in the environment. This information can be used in conjunction with some sort of map and/or localization technique, such as SLAM, to augment a navigation system and increase the usefulness of the system. Optical Flow is one of a number of methods which have been proposed to extract the apparent motion within an image sequence, but is one of the most extensively studied. Recovering image motion has many ther important applications, in fields such as video compression, where it is an essential component of the MPEG encoding process.

The origins of optical flow have been attributed to the work of Fennema and Thompson, though the term was first defined by Horn and Schunck [5] as the distribution of apparent velocities of movement of Brightness patterns within an image, based upon the apparent motion of regions of similar intensity over an image sequence. In its simplest form, this can be expressed as

$$\frac{dI}{dt} = \frac{\partial I}{\partial x}\frac{dx}{dt} + \frac{\partial I}{\partial y}\frac{dy}{dt} + \frac{\partial I}{\partial t}$$

To recover the optical flow from a sequence of images, the vector field of this motion, *o* (*x; y*) must be recovered from the intensity field *I*(*x; y; t*). A sample of visual odometry illustrates in Fig20.

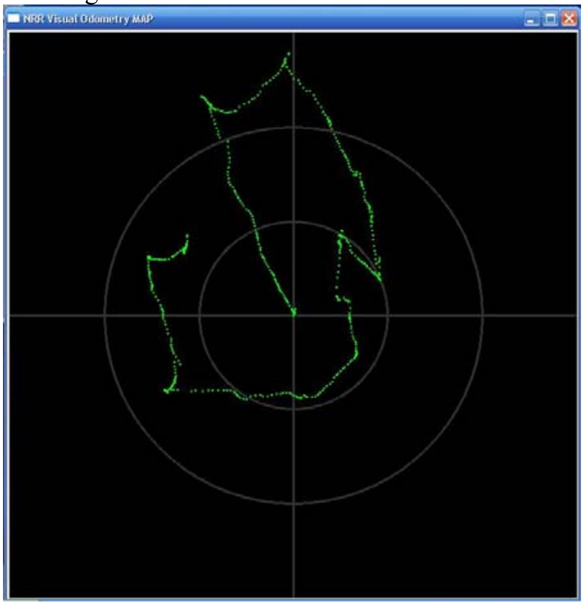


Fig 20.1 Visual Odometry 2D Map

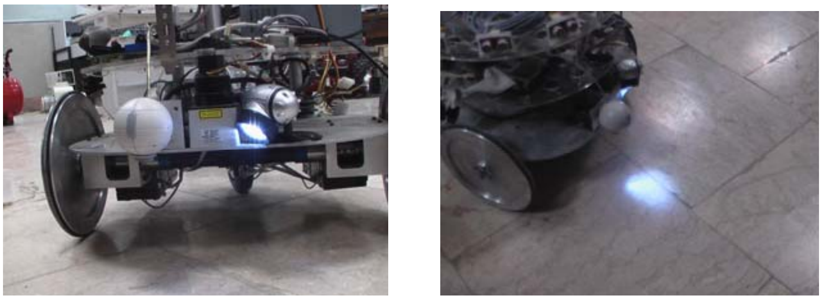


Fig20.2.(a sample of visual odometry)

11. Sound detection

In advance mobility robot A stereo microphone mounted with the cameras will be used to assist in the identification of the state of victims, the sound from the microphones will be fed back to stereo headphones worn by the operator. In autonomous robot, we perform audio-based victim detection by positioning two omni directional microphones with known distance. Given an audio source left, right or between both microphones, we are measuring the time difference between both signals. This is carried out by the Interaural Time Difference (ITD) and Cross-Correlation approach, which allows calculating the time delay of both signals based on the Cross-Correlation. As shown in Fig21, the bearing of the sound source can be successfully determined, even for different kinds of noise. [13, 14, 15, 16, 17]

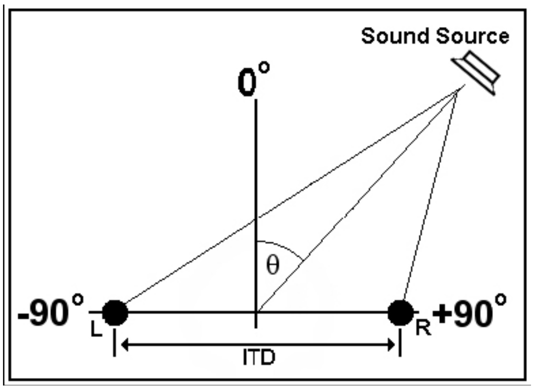


Fig21. The angle calculated to determine the azimuth. Left of 0_o is negative, right of 0_o is positive.

4. EXPLORATION & PATH PLANNING

Information can come in many forms, from knowledge of new terrain, to rock geology, to life forms. The value of these different information forms an explorer is determined by a set of information metrics, one for each form of information, that depend on the goal of the exploration task. As explorations become more complex, increasing numbers of information metrics must be considered in order to succeed. These multiple information metrics must be considered simultaneously during exploration and often conflict with each other to compete for the finite resources of the explorer. We can find different type of path planning algorithm in robotic field. In this project we used Generalized Voronoi Diagram (GVD) .

4.2 Roadmap Based Path Planning

Road Maps assume that global knowledge of the environment is available. They are the most common strategy for computing pre-planned paths i.e., the first step towards goal directed path planning. Usually, the set of paths are stored as a graph of nodes and edges

1 a raster grid (graph is implied by cell arrangement). The graph is pre-computed ahead of time without knowledge of start/goal locations. Start and goal locations are given later as a query. Often, a few additional edges are added to the graph here .In all cases, the graph is searched to find an efficient (e.g., shortest) path to the goal. Dijkstra's algorithm, A* or something similar use to find path between start and end. This methods categorized into two main categories:

- Sampling-Based Road Maps: divided in three parts
    1. Grid Based Sampling
    2. Probabilistic Road Maps
    3. Rapidly Exploring Random Tree Maps

- Geometry-Based Road Maps: that divided in three part
    1. Visibility Graph Paths
    2. Generalized Voronoi Diagram Paths
    3. Cell Decomposition Paths

4.2.2 Voronoi Diagram Implementation

We can find multiple ways of computing a Voronoi Diagram [50]. A Voronoi road map is a set of paths in an environment that represent maximum clearance between obstacles. They are sometimes preferred in robotics since they reduce the chance of collisions because sensors are often inaccurate and prone to error. Other names for this roadmap are generalized Voronoi diagram and retraction method. It is considered as a generalization of the Voronoi diagram for points. The Voronoi diagram for a collection of given points (called sites) is the graph formed by the boundaries of specially-constructed cells. Each of these cells surrounds one of the given sites and has the property that all points within the cell are closer to the enclosed site than to any other site. Voronoi diagrams can be generalized to situations in which the given sites are two-dimensional obstacles rather than mere points. In this type of problem, the boundaries of the specially-constructed cells are equidistant between the two nearest obstacles.

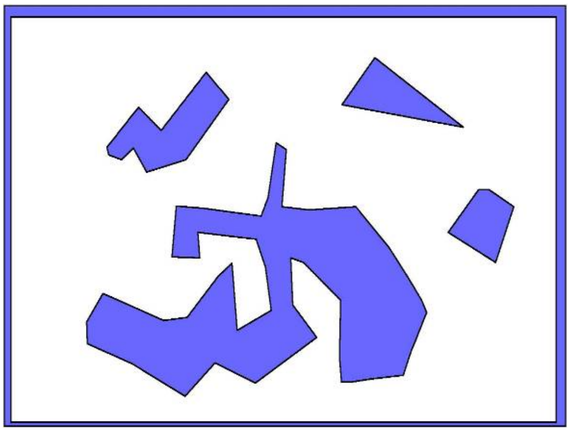


Fig 22 shows a simple sample of Voronoi diagram.

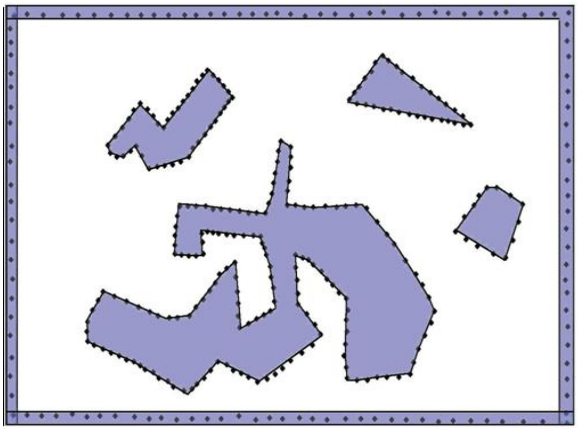


Fig 22: Part A: Compute sample points along obstacle border.

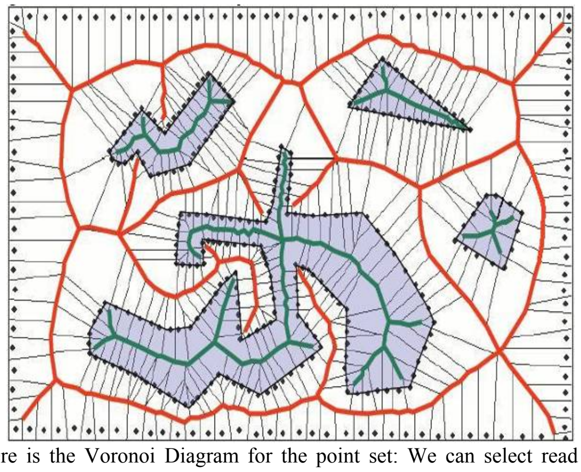


Fig 22: Part B: Here is the Voronoi Diagram for the point set: We can select read lines that show all possible paths in the environment. Resulting GVD edges can be searched for a path from start to goal (e.g., store GVD as graph, run Dijkstra's shortest path algorithm)

4.2.3 Computing the Generalized Voronoi Diagram

Computing the Voronoi diagram for a collection of n given points (sites) in two dimensions is a well-known problem. Steven Fortune [51] presented a novel and extremely efficient method for the computation. By using a sweep-line algorithm, he was able to reduce the time complexity to order O(n log(n)).The obstacles encountered in robot path planning are actually extended two dimensional objects, not simple points. One therefore needs an appropriate generalization of the Voronoi diagram in order to deal with such obstacles. Although the usual Voronoi diagram for a discrete set of given points only contains edges that are straight line segments, the edges of a generalized Voronoi diagram will contain parabolic arcs as well as straight lines. Although possible, it is algorithmically difficult to generate such a Voronoi combination of parabolas and straight lines (linked together at point vertices). Even if one did produce this generalized Voronoi diagram exactly, one would still have to approximate each parabolic arc by a collection of small line segments so as to produce usable commands for the robot's motion. Okabe, Boots, and Sugihara [52] have suggested a very useful approximation method for finding the generalized Voronoi diagram for a collection of two-dimensional obstacles. The boundaries of the polygonal obstacles can be approximated by the large number of endpoints that result from subdividing each side of the original polygons into very small segments. In this way, one obtains a set of discrete points for which one can compute the ordinary Voronoi diagram. Okabe, Boots, and Sugihara suggest that one then eliminate each and every Voronoi edge that was generated by two points located on the same obstacle. The remaining edges presumably give a good approximation for the generalized Voronoi diagram determined by the original two-dimensional obstacles.

Once the generalized Voronoi diagram for a given set of obstacles has been computed, one must connect the robot's actual starting and stopping points to the diagram. In Fig 23 you can see our implementation results of GVD exploration algorithm.

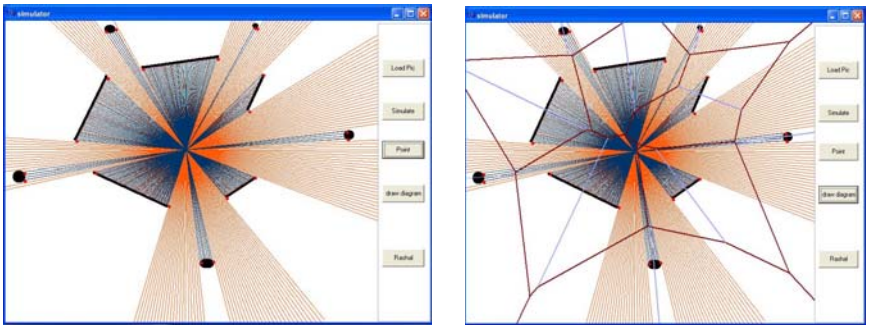


fig.23 voronoi algorithm path output

12.Robot Locomotion

We have designed two different types of robot for different arena (situation). But as a general rule all robots are based on differential servomechanism which is explained briefly. We have used two identical dc servomotors to drive each robot. The robot may travel forward or backward by driving two motors in CW or ACW direction and it may rotate around its center of gravity by driving motors in reverse directions.

It should be mentioned that all the mechanical design process are carried out by our team members using AutoCAD2002 and 3DWORKINGMODEL. All the robot parts are built by CNC machines.

The servomechanism control system is based on an embedded microcontroller and H-bridge driver using PWM method. The velocity and position of the servomechanism are controlled using PID control algorithm based on optical encoder feedback.

NAJI-II:

This robot is designed for outdoor and harsh environment. This robot is a new version of NAJI_V. Fig 24 illustrates the 3D views of the robot.

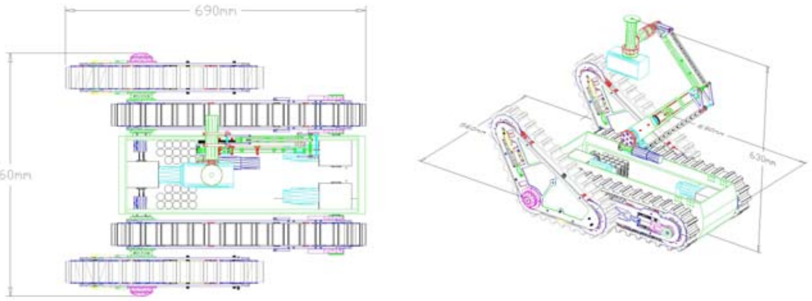


Fig 24: NAJI-V Dimensions.

It should be noted that we have also the robot NAJI-VII .at the time of preparing this TDP, the robot is designed and all individual parts are manufactured but not assembled yet. Fig 25 illustrates the 3D views of the robot.

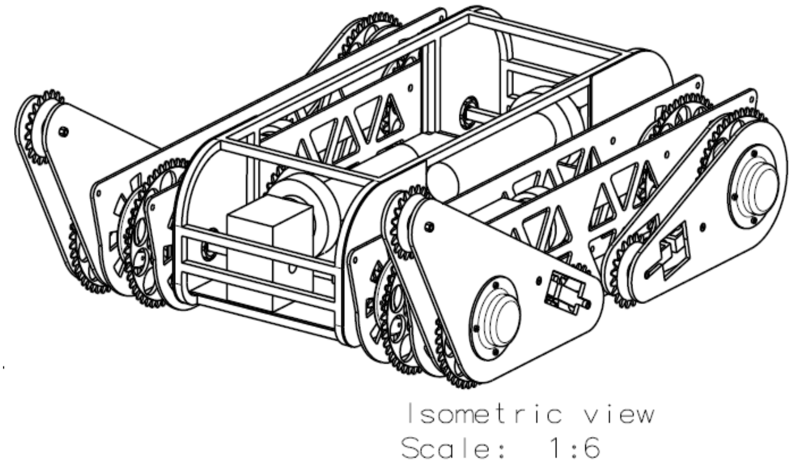


Fig. 25 NAJI_VII 3D view

Table 1: NAJI-V Specifications

Weight	30	Kg
Carriage Load	20	Kg
Max Velocity	47	Cm/Sec
Length	750	mm
Width	560	mm
Height	260	mm
Locomotion	2 DC servomotor	150 W

NAJI-IV:

This simple robot is just designed for indoor and flat surface. The control and navigating of the robot is quite easy. Dead-reckoning technique with digital compass and IMU, provides a very good and accurate localization method. The systematic mechanical errors compare to other NAJI robots is quite less. Fig.26 shows photo of NAJI-IV .

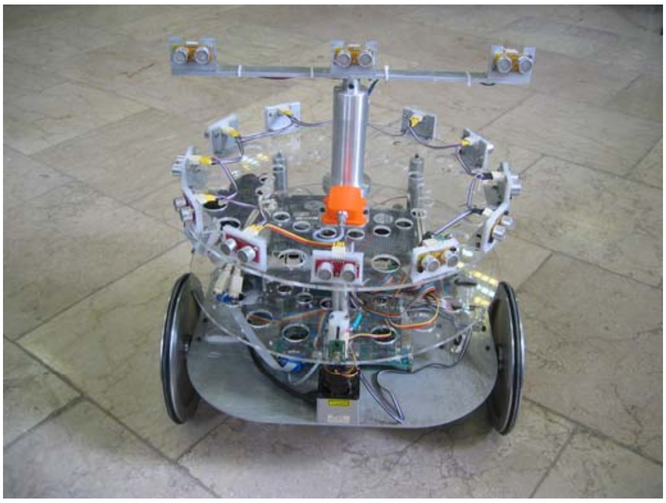


Fig .26 Photo of the NAJI-IV

Table 2: NAJI-IV Specifications

Weight	5	Kg
Carriage Load	7	Kg
Max Velocity	0.5	m/Sec
Length	378	mm
Width	378	mm
Height	180	mm
Locomotion	2 DC servomotor	120 W

13. Team Training for Operation (Human Factors)

As mentioned before we have managed to train two operators. We have also prepared a combination of the Yellow and Red arena next to the Mechatronics Research Laboratory to test our robots. So operators can improve their skill to command the rescue robot professionally.

14. Possibility for Practical Application to Real Disaster Site

As we experience so many earthquakes in our country, our main goal is not only to take part in Robocup competition but we are aiming to design and build real rescue robot for real situation. For example our first robot NAJI-I which was in Padova was tested in Bam earthquake to realize real problems and finding proper solutions.

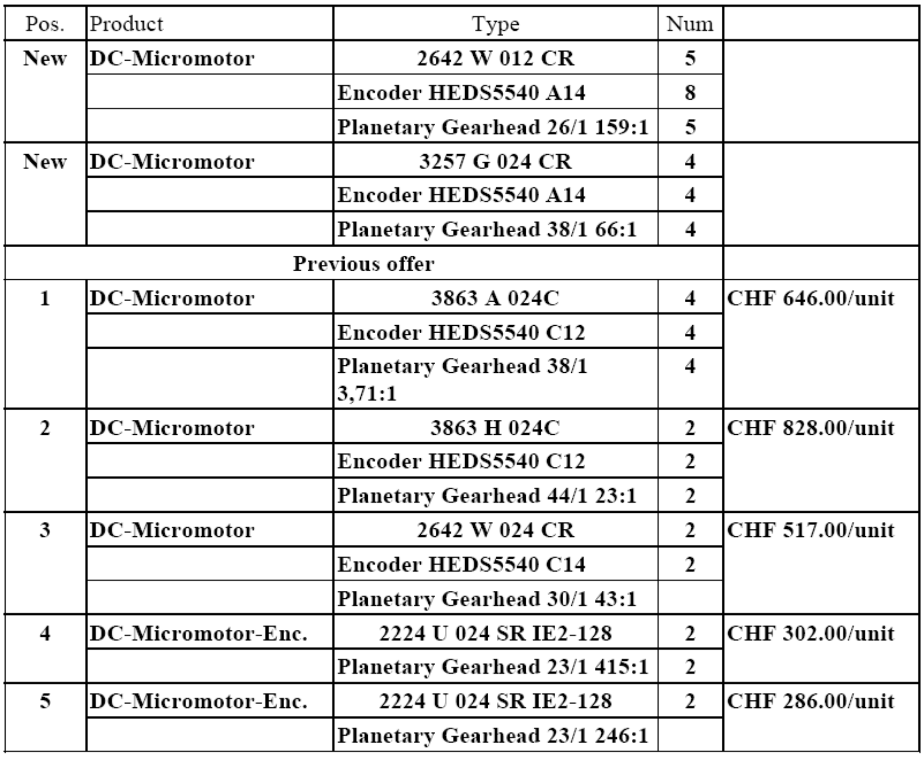
15. System Cost

Most of the mechanical parts of the robots are designed and built by our team members. Depends on the type of the motor we have bought our DC servomotors from Faulhaber. Ultrasonic sensors and digital compass are bought from Devantech Robot-Electronics. Other sensors, Wireless-LAN card and electronics parts are bought from local shop. The cost for each robot differs depending on the size and complexity of the robot but approximately each robot cost us about 5000-9000 $US.

Table 3. The sensor's cost:

Sensor's Type	Price Per Piece	Distributor
Co2 Detector	600 US$	www.edinst.com
Laser Scanner	180000 yen	www.hokuyo-aut.jp
Laser Scanner	4500 €	www.sick.com
Thermal Camera	14000 US$	www.flirthermography.com
Temperature sensor	43.02 £	www.robot-electronics.co.uk
Ultrasonic	25.50£	www.robot-electronics.co.uk
Camera 260x	329 US$	www.pinecomputer.com
IMU	2100 US$	www.xsense.com
Motors	*Table 4	www.minimotor.ch

Table 4. Motors Price List



Pos.	Product	Type	Num	
New	**DC-Micromotor**	**2642 W 012 CR**	**5**	
		Encoder HEDS5540 A14	**8**	
		Planetary Gearhead 26/1 159:1	**5**	
New	**DC-Micromotor**	**3257 G 024 CR**	**4**	
		Encoder HEDS5540 A14	**4**	
		Planetary Gearhead 38/1 66:1	**4**	
Previous offer				
1	**DC-Micromotor**	**3863 A 024C**	**4**	**CHF 646.00/unit**
		Encoder HEDS5540 C12	**4**	
		Planetary Gearhead 38/1 3,71:1	**4**	
2	**DC-Micromotor**	**3863 H 024C**	**2**	**CHF 828.00/unit**
		Encoder HEDS5540 C12	**2**	
		Planetary Gearhead 44/1 23:1	**2**	
3	**DC-Micromotor**	**2642 W 024 CR**	**2**	**CHF 517.00/unit**
		Encoder HEDS5540 C14	**2**	
		Planetary Gearhead 30/1 43:1		
4	**DC-Micromotor-Enc.**	**2224 U 024 SR IE2-128**	**2**	**CHF 302.00/unit**
		Planetary Gearhead 23/1 415:1	**2**	
5	**DC-Micromotor-Enc.**	**2224 U 024 SR IE2-128**	**2**	**CHF 286.00/unit**
		Planetary Gearhead 23/1 246:1		

References:

[1] Lauro Ojeda, Giulio Reina, and Johann Borenstein, “Experimental Results from FLEXnav: An Expert Rule-based Dead-reckoning System for Mars Rovers” University of Michigan Advanced Technologies Lab

[2] Xiaoying Kong, “Inertial Navigation System Algorithms for Low Cost IMU” Department of Mechanical and Mechatronic Engineering, The University of Sydney, August 27, 2000

[3] Krantz, D. and Gini, M., 1996, “Non-Uniform Dead-Reckoning Position Estimate Updates.” *Proc. of the 1996 IEEE Int. Conference on Robotics and Automation Minneapolis*, MN, April 22-25, pp. 2061-2066.

[4] *KEVIN J. WALCHKO, “LOW COST INERTIAL NAVIGATION: LEARNING TO INTEGRATE NOISE AND FIND YOUR WAY”, A THESIS PRESENTED TO THE GRADUATE SCHOOL OF THE UNIVERSITY OF FLORIDA IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF MASTER OF SCIENCE UNIVERSITY OF FLORIDA 2002*

[5] J. Andrade-Cetto, A. Sanfeliu, "Concurrent Map Building and Localization on Indoor Dynamic Environments", Institut de Rob`otica i Inform`atica Industrial, Spain.

[6] J. J. Leonard, "Directed Sonar Sensing for Mobile Robot Navigation", PhD Thesis, Cambridge, U.S.A, 2002.

[7] P. Holmberg, "Sensor Fusion with Coordinated Mobile Robots", Master Thesis Linköping University, Sweden, 2003.

[8] Ioannis Rekleitis, "Cooperative Localization and Multi-robot Exploration", Master Thesis, McGill University, 2003.

[9] V. Nguyen, A. Martinelli, N. Tomatis, R. Siegwart, "A Comparison of Line Extraction Algorithms using 2D Laser Rangefinder for Indoor Mobile Robotics", Autonomous Systems Laboratory Ecole Polytechnique F´ed´erale de Lausanne (EPFL), Switzerland.

[10] I. Hwang , H. Balakrishnan , K. Roy , J. Shin , L. G. C. Tomlin, "Multiple Target Tracking and Identity Management", Stanford University.

[11] Sunil Arya, Theocharis Malamatos, David M. Mount "Space Efficient Ap proximate Voronoi Diagrams" Appeared in STOC 2002, pp. 721-730.

[12] S. J. Fortune. A sweep line algorithm for Voronoi diagrams. In Algorithmic 2 (1987),Pages 153–174.

[13] Atsuyuki Okabe, Barry Boots, and Kokichi Sugihara. Spatial tessellations: concepts
and applications of Voronoi diagrams. John Wiley & Sons, Inc., 1992.

[14] J. Murray, H. Erwin and S. Wermter
Robotic Sound-Source Localization and Tracking Using Interaural Time Difference and Cross-Correlation

[15] D. Giuliani, M. Omologo, and P. Svaizer. Talker localization and speech recognition using a microphone array and a cross-powerspectrum phase analysis. In *Proc. ICSLP94*, pages 1243–1246, 1994.

[16] Hans-Joachim Böhme, Torsten Wilhelm, Jürgen Key, Carsten Schauer, Christof Schröter, Horst-Michael Groβ, Torsten Hempel.
An approach to multi-modal human-machine interaction for intelligent service robots. Robotics and Autonomous Systems, Vol 44(1) 2003, pages 83-96.http://www.google.com/

[17] www.akg.com
[18] www.edinst.com
[19] www.hokuyo-aut.jp
[20] www.robot-electronics.co.uk
[21] www.pinecomputer.com
[22] www.xsense.com
[23] www.minimotor.ch