

# On the Design and Development of A Rough Terrain Robot for Rescue Missions

J. Suthakorn\*, S.S.H. Shah , S. Jantarajit , W. Onprasert and W. Saensupo,  
S. Saeung, S. Nakdhamabhorn, V. Sa-Ing, and S. Reaungamornrat  
*Center for Biomedical and Robotics Technology (www.bartlab.org)*  
*Faculty of Engineering, Mahidol University, Salaya, Nakorn Pathom, Thailand*  
\*Corresponding Address: Email [egist@mahidol.ac.th](mailto:egist@mahidol.ac.th)

**Abstract** - Rescue Robots play an important role during rescue missions in disasters such as 9/11, which caused more than 2,000 deaths and thousands of injuries. However, tele-operating rescue robots are unable to perform their tasks constantly due to the limitation of current wireless communication technology. Therefore, rescue robots with the capability of performing their tasks autonomously during temporarily lost connections to the control base would be ideal. This paper introduces our development of a semi-autonomous rough terrain robot for rescue missions. The robot's hardware components, system architecture, and software architecture are described in order to provide a general overview of our robot. An alternative and comprehensive map-generating algorithm is presented and discussed. Finally, experimental setup and results from a testing arena are reported.

**Index Terms** – rescue robot, rough terrain robot, semi-autonomous robot, mobile robot, path planning, SLAM

## I. INTRODUCTION

A fully autonomous rescue robot with capabilities of self-navigating, victim searching and rescue-plan generating would be an ideal for everyone involved with search and rescue actions.

### A. The Statement of Problem

Disasters (both natural hazards and man-made catastrophes) have brought loss, grief, and starvation to survivors. An example of a deadly natural calamity was the Great Chilean Earthquake, which took place in Spain in May 1960; roughly 5,000 people died from both earthquake and resulting Tsunamis. In May 2008 more than 50,000 Chinese citizens lost their lives, and over 20,000 people were missing as a result of the massive devastative earthquake. Man-made disasters are also an important motivation, such as the grievous incident happening in USA on 11 September 2001 (9/11).

Most victims in 9/11 died due to the delay of assistance. In such conditions, the victims' locations and conditions were difficult to determine by rescue crews. Several researchers and academic staff, consequently, have paid more attention in conducting research to develop rough terrain robots, especially for rescue missions. Such rescue robots are able to perform their tasks in high-risk and dangerous places. The robots are able to supply images of the environment and specify victims' locations to the robot operators at the control base outside the wrecked area.

### B. Related Work

To develop a rescue robot, the key features worth considering are mobility and map generation. Sheh [1], for example, was interested in increasing the mobility of a robot. A toy "Tarantula" was modified to be a rescue robot called "The Redback." The robot had improved mobility because of its small size and light weight. S. Thrun and his co-workers [2] introduced the FastSLAM algorithm (A Factored Solution to the Simultaneous Localization and Mapping Problem) to help increase its accuracy in map generation.

Rescue robots sometimes work as a group robot. Vargas and his colleagues [3] constructed three cooperative mobile robots capable of detecting casualties in disasters. From another perspective, Birk and his colleagues [4] developed a rescue robot to examine a collapsing building in city areas and find victims. Vincent and Trentini [5] introduced a robot which was able to understand obstacle shapes and was able to climb over them rather than avoid them by applying image segmentation and shape detection algorithms.

In contrast with manually controlled robots, autonomous rescue robots must decide their travel paths in many circumstances which direct them to reach and gain victim information. This requirement leads several researchers to develop intelligent terrain

robots. Birk and Kenn [6], for instance, have developed a semi-autonomous rescue robot to overcome the limitation of wireless communication that impede constant control over the robot. In addition, Pellenz and his team [7] have developed a rescue robot to perform in either autonomous or tele-operating mode.

This paper presents the design and development of a rough terrain robot, “Tehzeeb” (Fig. 1). The content is divided into three parts: System Descriptions, Map Generation, and Experimental Results.



Fig. 1 The Tehzeeb Rescue Robot

## II. SYSTEM DESCRIPTIONS

This section presents the robot descriptions which can be separated into three parts; 1) Hardware Components, 2) System Architecture, and 3) high-level Software Architecture. The Tehzeeb rescue robot can be separated into four parts: front and rear arms, robot body, manipulators for carrying victim sensors and camera, and the electronic compartment (Fig. 2).

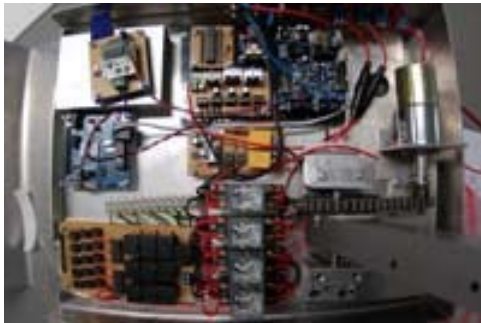


Fig. 2 The electronic compartment

Tehzeeb is a 30 kilogram mobile robot with a length of 97 centimeters (when both front and rear arms are stretched), 40 centimeters width, and eighty centimeters height including its manipulator. The front arms, themselves, are 40 centimeters long, while the rear arms are 27 centimeters in length. The radiuses of each arm is 270 centimeters. Tehzeeb is able to climb over

a 40 degree slope (Fig. 3). The robot navigates by using information from a laser scanner, creates maps, finds victims and sends information about the locations of casualties to the station via wireless communication (IEEE 802.11a).



Fig. 3 The Tehzeeb is on a 45 degree testing slope

### A. Hardware Components

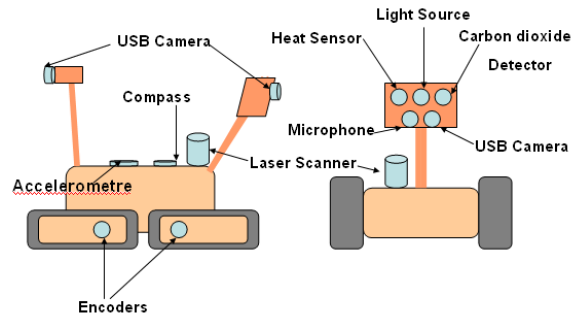


Fig. 4 Hardware Components

The Tehzeeb robot is equipped with several sensors, i.e. USB cameras, heat sensor, Carbon dioxide detector, and microphone. These sensors detect information about the victims. All of them are located on an onboard-manipulator. Fig. 4 illustrates the hardware components of the robot. There are various sensors that are responsible for tracking the robot position and orientation i.e. an accelerometer, a compass, and a laser scanner. The scanner is on a level-stabilizer to accurately measure the distances between robot and environments. The distance data is used to generate the surrounding map (detail is discussed in the following subsection.) The battery compartment is located at the center of the robot for better stability. A USB camera is used to capture rear images where the back arms could be seen through the camera whether they get stuck in the wreck. An encoder is employed to gain robot traveling data which is used in cooperate with the laser scanning data to generate a higher accuracy map.

### B. System Architecture

This subsection describes the system architecture of the Tehzeeb robot (Fig. 5). A PIC micro-controller is used to control and interact with the manipulator, heat sensors and carbon dioxide detectors, as a low-level controller. The PIC micro-control, then, communicates to the master controller, where an ARM7 controller is used. The ARM7 master controller is also responsible for controlling and interacting with several devices, such as, front and rear arms, robot driving system, accelerometer, electronic compass and encoders. The master controller, then, connects to an on-board laptop via USB port.

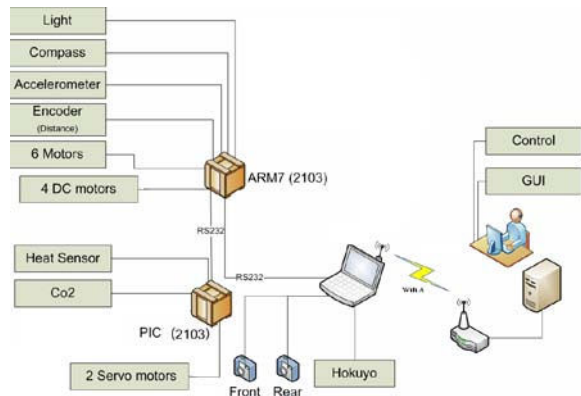


Fig.5 The robot's System Architecture

The laptop takes care of data processing gains from devices. The communication between the robot and control base is done through the laptop's wireless system. Table 1 lists the hardware devices which are used in the robotic system.

### C. High-Level Software Architecture

Fig. 6 illustrates the robot's software architecture. The software consists of seven packages: Hokuyo, Compass, Front Camera, Sensors, Robot State Information and Window Form Application. The main module that controls other classes is the Window Form Application module. This module communicates with the Map module to obtain data to draw a map and it also retrieves and keeps angles that the robot rotates from the Compass module. Besides this, the front camera position and direction is controlled by the Form module via the Camera module. After gaining all necessary information (data from all sensors, measurements from a compass, map data), the Window Form Application module will display all the information it attains on its display window.

TABLE I  
HARDWARE DEVICES

Components	Brand and Version
Servo Motor	GWS S666/STD
DC Motor	Tormax
Microcontroller	PIC 18F2331
	ARM 7 (LPC 2103)
Encoder	Yaskawa model 200 ASKS 5VM
Compass	ADX-CMPS03
Accelerometer	Parallax Memsic 2125
Motor	ZGB70-60SRZ-1
Heat Sensor	Thermopile Sensor SMTIR 990 2S1L
Carbon dioxide Detector	TGS 4161
Laser Range Finder	Hokuyo URG-04LX
USB Cameras	Logitech QuickCam Pro 9000
Note book	Compaq

Secondly, the Map module is responsible for communicating with the Hokuyo module (a laser range finder), which is an interface to send commands and receive responses from the Hokuyo laser scanner. Raw data obtained from the Hokuyo class is filtered and processed to gain the environment that is described in the world coordinate by this class before transferred to the main module and displayed.

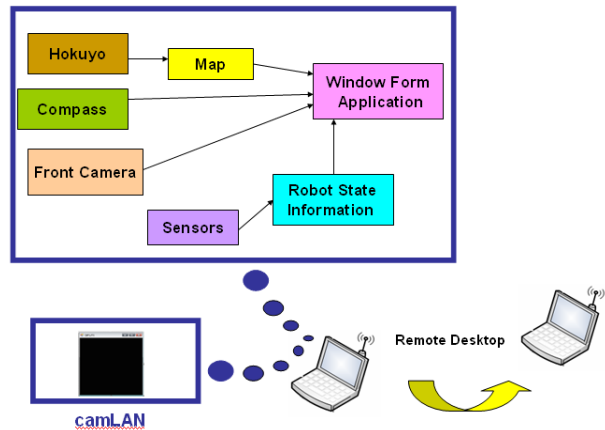


Fig. 6 The robot's Software Architecture

For the Compass module and the Sensors class, they have the same duty, which is to connect to their hardware and retrieve data to display on screen. However, the Camera is not responsible for display a stream video as many people might assume from its name; it only receives the order from the main module to control the position of the camera. For displaying the stream, we apply software and run them separately and concurrently. This is because it reduces time processing of our program. Finally, instead of sending data via arbitrary wireless communication and processing them at the remote station, we decide to process all data on

the on-board computer and apply a remote-desktop to demonstrate the laptop screen on the monitor of the computer station.

### III. MAP GENERATION

The information about distances and angles to obstacles obtained from the Hokuyo laser scanner is crucial for constructing maps. The strategy that we apply to complete this is alternative and comprehensive; it is comprised of an algorithm to select and recognize landmarks from each scan, a procedure to match landmarks to the previously collected ones, and a method to localize itself while exploring an unseen region.

#### A. Method and Algorithm

We utilize a simple and straightforward method to select landmarks from each scan as we define that landmarks are points which own the significant and outstanding features exposed in graphs plotted in polar coordinate; however, we have to filter data before apply this procedure because of the vibration of the scanner.

##### 1) Landmark Location Evaluation

Noticeable and different features are used to determine landmarks and match them to the existing ones (landmarks from the previous scan). Those features expose from plotting a graph in polar coordinates; graphs will be distinct according to the real environment.

*The first feature* is a relative maximum distance; we can find such points in the distance-step graph. This feature represents a corner or informs that there will be free space if after that peak, the graph drops to zero. (Fig 7. illustrates the robot travelling through a three junction. Fig. 8 and Fig.9 present graphs plotted in Polar and Cartesian coordinate to represent landmarks gaining from this relatively maximum distance feature.)

*The second feature* is obtained from a graph of steps and derivatives of distance by step multiplied by two. For this feature, a landmark is a point that has the relatively highest twice derivative value. This feature represents corners of real-world junctions if there are two corners around a junction, lasers scanner will find two points that have relatively highest double derivative value. (Fig 10. illustrates the robot travelling through a three junction in the different direction from Fig. 7. Fig. 11 and Fig. 12 depict graphs plotted in

Polar and Cartesian coordinate to represent landmarks gaining from this relatively maximum twice derivative value feature.)

$$x = \text{step}, y = \text{distance}$$

$$f(x) = 2 \times \frac{dy}{dx}$$

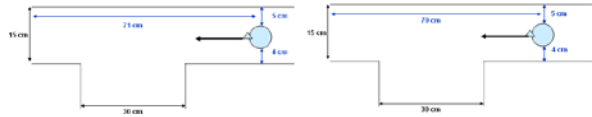


Fig. 7 The robot moves straight forwards through the three junctions

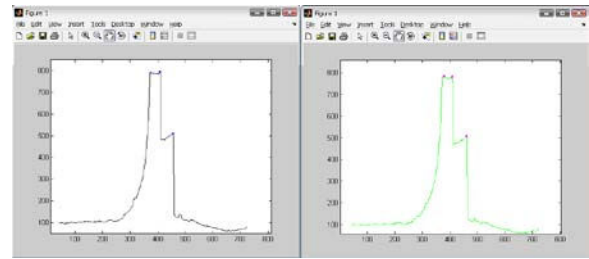


Fig. 8 A graph plotting in the Polar Coordinate representing landmarks

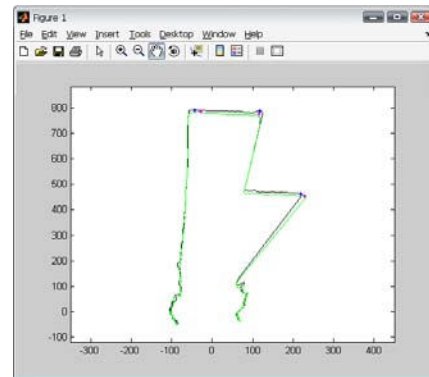


Fig. 9 A graph plotting in the Cartesian coordinate

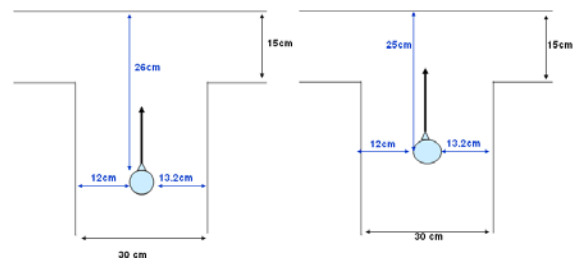


Fig. 10 The robot moves straight forwards through the three junctions



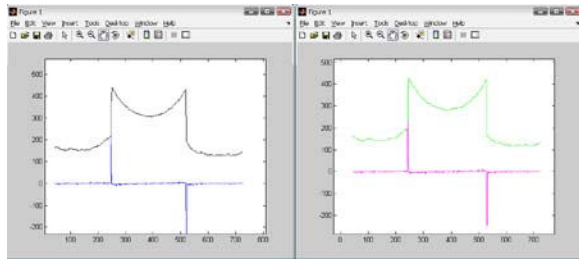


Fig. 11 Graphs of distances and steps plotted in the Polar coordinate and graphs of double derivatives and steps displaying landmarks at their peaks.

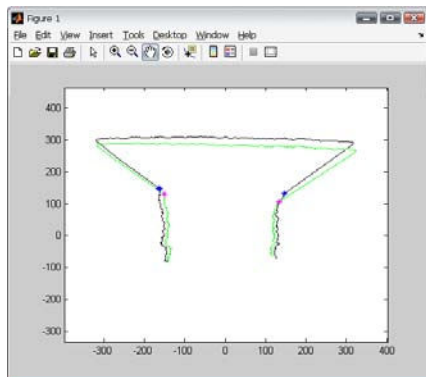


Fig. 12 Graphs of distances and steps plotted in the Cartesian landmarks at their corners.

These two features will be applied to select landmarks from each scan and used to match landmarks from the previous scan to the current one. We, firstly, have to check whether the sequences of the landmark features obtained from both rounds is the same, if so, then we check whether gaps between distances of the previously-scanned step and of the currently-scanned step is acceptable. If it is not too large, those points are located in the same positions in the world coordinate.

After we achieve matched-landmark pairs, we will apply it to figure the translation and the rotation of the robot by utilizing geometry theory and vector property; then the translation vector will be used to calculate the angle of rotation by solving the transformation equation which converts the currently-scanned landmark to the similar previously-scanned one. With this information (the position vector, and the angle of rotation), we then can form a homogeneous transformation matrixes which will use to modify the local description of environment to the world coordinate. Ultimately, with the environment described in the world coordinate, we can complete plotting a map.

#### IV. EXPERIMENTAL RESULTS

Our robot was demonstrated and we observed its performances in five perspectives during participating in a robot competition, “Thailand Rescue Robot Championship 2007” (the arena for this competition was presented in the Fig. 13.) The robust mechanical structure and mobility of the robot was examined by letting it travel along various paths: a rough and dangerous step field, which is risky for the robot to break and out of order (as illustrated in Fig. 14), and the smoother region. While the robot was surveying, the environmental information was used to determine path and to search for victims, so cameras and sensors were tested. The software for robot’s control, robot’s status display, and map generation were monitored. The experiment indicated that the mechanical structure of our robot was not robust enough in the severely uneven environment, so material and design were necessary to be changed and improved. The software for map generation was also incomplete and essential to be added a module to cope with the error accumulation.



Fig. 13 The arena for Robocop Rescue Robot League

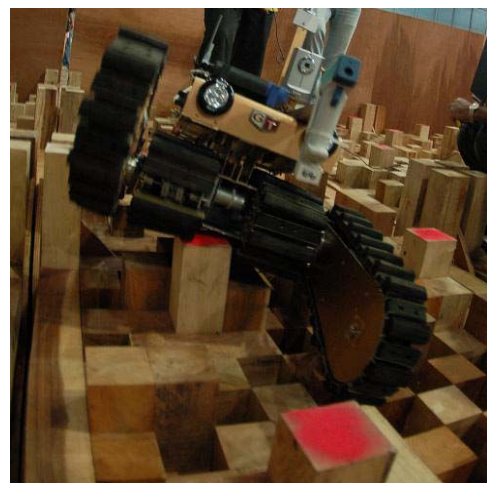


Fig. 14 the robot on random step field

## V. DISCUSSION

To be a complete mobile rough terrain robot, we have to modify our robot mainly in two major parts. Firstly, its mechanic designs and implementation have to adjust to make it more robust in difficult environment. Secondly, the software that is responsible for map-generating and path-planning have to improve and add other procedures to make the robot become truly autonomous.

## REFERENCES

- [1] R. Sheh, "The Redback: A Low-Cost Advanced Mobility Robot," the University of New South Wales, Sedney, Australia
- [2] M. Montemerlo, S. Thurn, D. Koeler, and, B. Wegbreit, "FastSLAM: A Factored Solution to the Simultaneous Localization and Mapping Problem," Proceedings of the AAAI National Conference on Artificial Intelligence, 2002.
- [3] A. E. M. Vargas, K. Mizuuchi, D. Endo, E. Rohmer, K. Nagatani, and, K. Yoshida, "Development of a Networked Robotic System for Disaster Mitigation –Navigation System Based on 3D Geometry Acquisition-, " Tohoku University, Japan
- [4] A. Birk , H. Kenn, S. Carpin , and, M. Pfingsthorn, "Toward Autonomous Rescue Robots," First International Workshop on Synthetic Simulation and Robotics to Migrate Earthquake Disaster, July 5<sup>th</sup>, 2003.
- [5] I. Vincent and M. Trentini, "Shape-shifting Tracked Robotic Vehicle for complex terrain navigation," Defence R&D Canada, Technical Memorandum, DRDC Suffield TM 2007-190, December 2007
- [6] A. Birk , H. Kenn, "A Rescue Robot Control Architecture ensuring Safe Semi-Autonomous Operation," , {Robocup}-02: Robot Soccer World CupVI, Springer, INIST-CNRS, Cote INIST, February, 2004
- [7] J. Pellenz, "RoboCup 2008- RoboCupRescue Team resko@UniKoblenz", University of Koblenz-Landua, Germany, 2008