THE UNIVERSITY OF HULL

# Navigation in unknown environment by building instantaneous spatial structures

being a Thesis submitted for the Degree of PhD

in the University of Hull

by

N.HU, BSc

January, 2011

# ACKNOWLEDGMENTS

***There were lots of many significant things happened to my family and myself during these years of my PhD research.***

***Great thanks to my supervisor Dr. Chandrasekhar Kambhampati for his support and looking after me during these years.***

***Great thanks to my family and all my best friends for their trust in me and support during these years.***

# TABLE OF CONTENTS

# LIST OF FIGURES

# LIST OF TABLES

# ABSTRACT

A strategy typically employed for mobile robot navigation in an unknown environment is to follow a nominal straight-line path to the goal point. During travelling on the nominal path, the robot uses distance information, e.g. derived from sonar sensors, and geometric information to determine the spatial relations between the robot and the environment. Navigation in an unknown environment is still a challenging issue especially in the presence of cluttered objects or obstructions.

There are two possible ways to path planning in an unknown environment: the first is to map the environment and navigate based on the map; the second is to assign a nominal path, which the robot follows whilst at the same time it senses obstacles and reacts to achieve a collision free trajectory. In both cases the robot circumnavigates obstructions and generates a new path from the initial location to the goal point. Often the strategies used for navigation employ simple path planning techniques aided by specific methods to recognize objects and construct a structure for the environment. In Chronis' PhD thesis is this area, a ring of low level sonar sensors is used to get spatial relations between a mobile robot and its environment. The eventual goal is to use spatial relations for navigation of the mobile robot in an unstructured, unknown environment. However, Chronis' work does not construct any model of perceived structures in the environment and does not involve any tolerance to sensor failure. The approach described in this thesis improves this earlier work in precisely these two areas.

The proposed approach uses low level sensors, such as sonar sensors, to achieve navigation in an unknown and cluttered environment. It integrates sonar sensors and geometric information to construct structures of the environment and consequently establish a system that navigates effectively and quickly through cluttered objects and obstructions. It is shown that this strategy achieves efficiency and effectiveness in mobile robot navigation. The approach is also shown to be robust and tolerant to sensor failures. The strategy is not dependent on the number or type of sensors on the robot and does not assume a particular type of robot; it can work with any sensory method that can provide an object representation in two dimensions.

# Chapter 1 Mobile Robot Navigation in an Unknown Environment

## 1.1 Background and Motivation

The issue of navigation raises three questions: “where am I?”, “where am I going?”, and “how should I get there?” It essentially consists of two problems, (a) the problem of localisation arises from the question “where am I?” and deals with the issue of determining the position in a particular environment [Kayton (1989), Rau (2003)], and (b) on the other hand, the problems of path planning and path following arise from the questions of “where am I going?”, and “how should I get there?” respectively (see Chapter 2 for more details in navigation). Consider the case of a mobile robot, moving from a Point (**S**) to another point (**G**). The robot has to plan a path from **S** to G by avoiding obstacles and possibly also satisfying other constraints such as optimizing the time taken, and utilizing minimal amount of energy. In order to achieve the above mentioned goals, the robot has to (a) plan a path and (b) travel along this path to the goal point. In other words, the path planning problem can be divided into global path planning and local path planning which is essentially obstacle avoidance. The problem of path planning can be categorized into a) path planning in a known environment and b) planning a path in an unknown environment.

In a known environment, global path planning requires the knowledge of the environment, and the planning algorithm generates a set of nominal points by which the robot will pass. A local navigation system executes the same steps as in the global path planning, by comparing the robot’s current position with positions stored in the global model, planning local paths as needed and avoiding unexpected obstacles. The local navigation system therefore must be made aware of the continuous changes in the environment by the incoming data gathered from sensors. The purpose of local obstacle avoidance is to plan a safe path which will take the robot around unexpected obstruction. In the known environment, the information of the world is provided a priori, e.g. a) the geometric features of the world are provided to the robot a *priori*; and b) the shape and location of obstacles are known to the robot. A collision free path can be found off-line, and followed online. The problem of path planning in a known environment has been actively and extensively researched [see e.g. Latombe (1991), Tan (2006), Duan (2004), Bruce & Veloso (2002), Weng (2005)].

However, it is the case that a robot will face a more general problem, where it has partial information or in the worst case no information of the entire environment. Since the robot has no pre-knowledge of the locations of the obstacles, it is not possible to plan a priori collision free path. There are two possible ways to path planning in an unknown environment: a) maps the environment and plan path based on the map, and b) assigns a nominal path, the robot follows the path and at the same time the robot senses and reacts for a collision free trajectory. The nominal path could be a straight path between start and goal points and the mobile robot will make online decisions to achieve the goal point and avoid collision along the way. A strategy often used on encountering an obstacle is to approach it, and then circumnavigate it by using the currently available information**.** Thus while navigating around the obstacle, the robot, at every instant, determines whether there is a path towards the goal. When such a path becomes feasible, the robot leaves obstacle and starts to move towards the goal once again. In such situations, the robot must gather information by using its sensors to explore the environment while moving and modifying its plans accordingly. Most navigation algorithms for an unknown environment do not attempt to optimise the length of the path because the safe circumnavigation is more concerned than the construction of a minimum length path. This is in contrast to the navigation in a known environment where path planning can be essentially reduced to one of finding optimal path in the given environment. However the path planning in an unknown environment also concerns with keeping computational overhead low, and achieving greater efficiency, robustness and fault tolerance ability.

The answer to the question of "where am I?" is generally the robot determines coordinates of its current position. In an unknown environment this is often not the completed answer. This is because this requires further information, for example the robot is in a corridor, corner or facing a front wall. The answers to the questions “where am I going?”, and “how should I get there?” in an unknown environment are given by the desired goal point, and a nominal path respectively. Thus in an unknown environment, when the robot meets an obstacle, it could circumnavigate it and generate another nominal path from there to the goal point. This could lead to the associated problem of minimizing deviations from a nominal path.

There have been a number of studies in this area [Wang (2004), Saffiotti (2000), Ye (2000), Seraji (2002), Arkin (1998), Dudek (2000), C.Ye (2009), J.Ng (2010)]. Often the strategies employ simple path planning but have specific methods to recognize objects and develop a structure for the environment. Wijk and Christensen [Wijk (2000)] provided a solution for the cluttered environment, using data obtained from sonar sensors. However, their method is deterministic in the sense that they did not incorporate any uncertainty in the detection of edges of objects within the vicinity of the robots themselves and in addition it does not incorporate any element of tolerance to sensor failure [Wijk (2000), Kambhampati, (2003)]. Chronis’ work [Chronis (2002, 2007)] provides a different perspective for path planning. It shows how linguistic expressions can be generated to describe the spatial relations between a mobile robot and its environment, using readings from a ring of sonar sensors. The eventual goal is to use low level sensors such as sonar, that generates linguistic description for navigation of the mobile robot in an unstructured, unknown, and possibly dynamic environment. It generates linguistic descriptions that represent the qualitative state of the robot with respect to its environment, in terms of which are easily understood by human users. Thus an exact model of the environment is not built, but an approximation of the local environment is generated. The robot used by Chronis is a Nomad 200 robot with 16 sonar sensors evenly distributed around the robot. The sensors’ readings are used to build an approximate representation of the objects surrounding the robot. During detection sonar sensor returns a range value which is less than the maximum, indicating that an obstacle has been detected. On the other hand when sonar sensors return the maximum value, it means that no obstacle has been detected. However the depth of the obstacle cannot be determined from the sonar reading. In the case of multiple sonar returns, a question arises as to whether adjacent sonar readings are from a single obstacle or multiple obstacles. Chronis' solution [Chronis (2002)] is to determine if the robot can fit between the points of two adjacent sonar returns. If the robot cannot fit between two returns, then they consider these returns to be from the same object. Even if there are actually two objects, they may be considered as one for the purposes of navigation. In the case when the distance between the two points of the sonar returns is big enough to allow the robot to travel through, the method considers this situation as separate objects. For example, this linguistic approach is illustrated by Figure 1.1 and Table 1.1. Object 1 is on the left of robot. The obstacles behind the robot are recognized as a single object (Object 2). The obstacle to the right of the robot is detected as three different objects. Since there are only three sonar readings from the right obstacle, and they are far apart according to the distance measure, the readings may not be from a single obstacle.

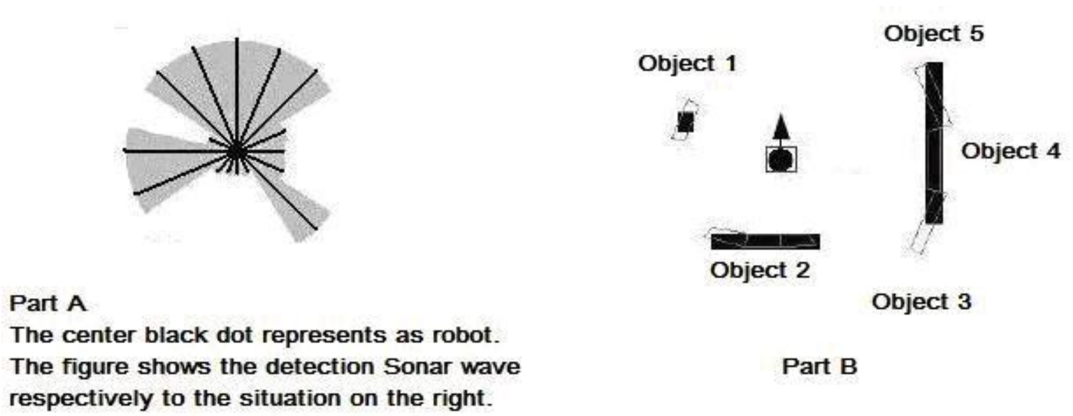


Figure 1.1a The sonar detection the robot in the situation of Part B;

Figure 1.1b, The robot detects 5 objects.

| |
|---|
| *"**Object 1** is mostly to the left of the Robot but somewhat forward "*<br><br>*"**Object 2** is behind the Robot but extends to the left relative to the Robot "*<br><br>*"**Object 3** is mostly to the right of the Robot but somewhat to the rear"*<br><br>*"**Object 4** is to right of the Robot"*<br><br>*"**Object 5** is mostly to the right of the Robot but somewhat forward "* |

Table 1.1 shows the linguistic description for Figure 1.1.

There are other methods similar to Chronis approach are developed [see Gribble et al (1998), Perzanowski et al (1999), Shibata (1996), Stopp (1994), Freeman (1975), Bloch (1999), Miyajima and Ralescu, (1994)].

## 1.2 Research Hypothesis, Thesis Aims and Objectives

Problem of path planning in an unknown environment is complex, computationally intensive and involves uncertainties of many kinds e.g. sensors, environments etc. Although the approach in this thesis is designed to use low level sensors, such as sonar sensor, to achieve the navigation in an unknown and cluttered environment; that integrates sonar sensors and geometric information to construct structures; that achieve more efficiency which consequently to establish a navigation system with quicker response, more effectiveness and more successful runs. As a result, the common uncertainties of sonar sensors are angular uncertainty and specular return, for the robustness of the approach that in order to tolerant these uncertainties and in addition the fault tolerance to sensor failures also becomes a feature for this approach.

The thesis will address following issues:

- In comparison with Chronis' approach which does not build any structures of environment and only gives the rough location of the detected object, can more sense be made about the environment based on homogenous sonar sensors?

- The approach must work with cluttered and irregular shaped objects while Chronis' approach is more like survivals in a structured environment.

- What happens in the event of a sonar sensor failure? Would the robot be capable of navigating safely to the goal point?

Thus the aim of this thesis is to develop path planning strategy for a mobile robot that would enable it to move from a starting point to goal point. It can gather information from environment based on a set of homogenous sonar sensors, in order to provide a structured view of the environment rather than rough locations. It should perform the task of reaching the goal point in a cluttered environment without collision, and even in event of sensor failure(s).

This research is aimed to achieve the following objectives:

- To develop and analyse an algorithm that can use sonar information and provide more information regarding the environment, in that it should classify environment into structures.

- To develop a path planning approach that can navigate the robot from a given starting point to a goal point, while using the structural information represent environment.

- To investigate the algorithm for fault tolerant capabilities e.g. when sonar fails.

- To set up experiments to verify the algorithm this provides structural classification.

- To set up experiments to test path planning approach and to compare with existing approaches, e.g. the the approach of Chronis.

### 1.3 The Structure of the Thesis

The algorithm in this thesis is implemented, verified and tested on the Saphira simulation platform. In Chapter 2, the approaches to navigation and path planning are reviewed. As the robot moves around, it gathers information by using sonar sensor. The information is used to classify the environment into a structure view. In order to do this, an algorithm is developed which is described in Chapter 3. There are uncertainties associated with sonar sensors, e.g. sonar specular return which causes wrong reading and results a bad classification or a poor path planning. This then leads to the issue of robustness of the approach. The algorithm should be able to carry on working in the presence of poor sensor information and failure of sensors. Chapter 4 deals with this particular aspect. Results are presented, The test results and verifications of the classification rules will be shown in Chapter 5 and the local avoidance strategies for each structure are introduced in Chapter 6. Chapter 7 reports the experiments and results of this project. The conclusion and future work can be found in Chapter 8.

# Chapter 2 The Fundamentals of Mobile Robot Navigation

## 2.1 Introduction

Robots are growing in complexity and their use in industry is becoming more widespread. The main use of robots has so far been in the automation of mass production industries, where the same, well defined tasks must be performed repeatedly in exactly the same fashion. However this project deals with the issue of navigation for a mobile robot. The mobile robot can be used in many areas, for example, office delivery, providing tours to museum visitors, exploring an unknown environment [C.Ye (2009), J.Ng (2010)] and etc. A mobile robot should be able to move in an environment with little or no human interference and it should be able to sense and react to its environment intelligently [Alonzo (1996)]. Although the robots seen in the science fiction movies appear to navigate with precision, in reality mobile robot navigation is a difficult research problem. Indeed, to simply let robot to navigate itself in a truly autonomous fashion is a serious challenge for today's mobile robots. Robot navigation is defined as the guiding of a mobile robot to a desired destination or along a desired path in an environment characterized by terrain and a set of distinct objects, such as obstacles and landmarks [Cao (1999)].

In this chapter the basic issues of mobile robot navigation is discussed. The problem of navigation essentially consists of the following questions:

- "Where am I?" which is essentially the localisation problem.
- "Where am I going?" and "How can I get there?" is the path planning problem including online decision path plan and obstacles avoidance.

The solution to localisation problem allows the robot to know where it is. Later on in this chapter the problem associated with localisation are discussed. The localisation of mobile robot can be classified into two categories: (a) reference based localisation, also called absolute position measurements, e.g. landmark navigation, active beacons and etc; and (b) dead-reckoning, also called relative position measurements which include odometry, and inertial navigation.

The second issue of navigation is path planning. The approaches to path planning can be broadly classified into two categories: model based path planning, whereby complete information on the environment is available *a priori* and the task is to find a collision-free path, and non-model based path planning, whereby the robot has to rely on its sensors for obtaining information on the environment. These approaches will be briefly reviewed in section 2.4 of this chapter.

## 2.2 The Issue of "Where I Am".

The issue of "where I am" is crucial for the robot because it allows the robot to localize itself; in other words position itself in the environment. This would then allow it to navigate the environment efficiently. Dead-reckoning is the process of determining current position by using course, speed, time and distance to be traveled, e.g. using odometry. The absolute positioning techniques, for example landmark navigation, use the landmarks' coordinates and shapes to determine robot's position. As the robot finds these landmarks, its position can be calculated. Due to the lack of a single good method, developers of mobile robots usually combine two methods, one from each group [Borenstein (1996)]. The two groups can be further divided into the following seven categories:

I: Relative Position Measurements (also called Dead-reckoning)

1. Odometry
2. Inertial Navigation

II: Absolute Position Measurements (Reference-based systems)

3. Magnetic Compasses
4. Active Beacons
5. Global Positioning Systems
6. Landmark Navigation
7. Model Matching

Odometry is the most widely used navigation method for positioning a mobile robot, which has advantages of short-term accuracy, cost-efficiency and high sampling rates. However, it also has an innate disadvantage, an accumulation of a variety of errors, which might result from the integration of incremental motion information over time. For example one type of error, orientation error will cause large positioning errors, which can increase proportionally according to the distance covered by the robot (Borenstein 1996). Borenstein and his colleagues (1996) categorized errors into two kinds: systematic and non-systematic errors. Systematic errors are those originating from kinematic imperfections of the robot, for example, unequal wheel diameters or uncertainty about the exact wheelbase. Non-systematic errors are coming from the interaction of the floor with the wheels, for example, wheel slippage or bumps and cracks. As a result, Borenstein (1996) and his colleagues developed a method, UMBmark for quantitatively measuring systematic odometry errors, to measure non-systematic errors to some extent. Borenstein and Feng (1995) also proposed a method for measuring non-systematic errors, called Extended UMBmark, which can be used for comparison of different robots under similar conditions. Although as far as the measurement of non-systematic errors is concerned, there are restrictions since it is heavily dependent on the floor characteristics, although its merit cannot be denied. Owing to a system of well-defined floor irregularities, in response to non-systematic errors, the UMBmark procedure can give susceptibility to a different-drive platform. See more details from Borenstein and Feng (1994, 1995, and 1996). In this project the Sappire Simulation System includes errors concerning realistic simulation

## 2.3 The Issue of "Where I Am Going" and "How I Can Get There".

The questions of "where I am going?" and "how I can get there?" are generally the genesis of a plan, i.e. a path to a specified goal and the ability to execute this plan and to modify the plan as necessary to avoid unexpected obstacles [Crowley (1985)], being called path planning. It is important to ensure that the robot avoids collision. The path planning problem can be divided into global path planning and local obstacle avoidance (obstacle avoidance unit). Global path planning requires the pre-knowledge of the environment, and the planning algorithm generates a set of nominal points which schedules the robot actions. A local navigation system executes the steps in the global plan, compares the robot's current position with the positions stored in the global model and plans local paths as needed to avoid unexpected obstacles. The local navigation system therefore must be made aware of the continuous changes in the environment by the incoming data from different sensors located at strategic positions onboard. Sensor fusion requires a good knowledge of each detector's response function to provide accurate resolution of systematic errors that result from inappropriate interpretation of the sensor data. The purpose of local obstacle avoidance is to plan a safe path which will take the robot around unexpected obstacles.

The navigational task of a robot cart moving on a flat floor of an obstacle-filled room is the subject of the work by Cahn [Cahn (1975)]. Starting from an initial vehicle location, the robot is to move to an externally specified new location or goal as directly as possible without colliding with any stationary or moving obstacles. The obstacles on the nominal path can be detected by sensors. The robot generates the obstacles avoidance path, and modifies the speed, turning angle etc. Within the spatial constraints imposed by the obstacles, this approach assumes the spatial constraints to be a simple structure such as a wall, a corner and etc; the trajectory must be formed and the robot moves along it until the robot comes back to the original nominal path. In general robot path planning in a known environment can be defined as:

Given a description of the environment, the robot plans a path from starting point to the goal point without collision. The solution path between the starting point and goal point must collision free and allow some clearance space between the obstacles and the robot. The solution path should be the shortest or least computation one among possible paths between starting point and goal point.

Path planning in an unknown environment has to be performed based on the information obtained from the sensors, which usually consists of the radial range and the azimuth. The radial range which is the distances of the obstacles from the robot and the azimuth which is the angle between the radial range and one the fixed axes of the robot's reference system. In the known environment, the main concern for path planning is to find an optimal path in the given environment. Navigation algorithm for an unknown environment does attempt to optimise the length of the path because the safe circumnavigation is more of an overriding concern than the construction of a minimum length path at the same time concern of computational overhead, related articles [C.Ye (2009), J.Ng (2010) ].

## 2.4 Navigating the Environment

Navigation has become a subject of significant interests in the modern world because it is open to a broad range of possible applications. Solutions of path planning algorithm have been proposed since the 1970s. A path planning algorithm among polyhedral obstacles based on the geometry graph was proposed in 1979 [Lozano-Perez (1979)]. Lumelsky and Stepanov [Lumelsky and Stepanov (1987)] considered the case where the automaton is a point and the environment is a subset of the two dimensional plane. The environment is assumed to be filled with unknown obstacles of an arbitrary shape and size. The information about the obstacles comes from a simple sensor whose capability is limited to detect an obstacle only when the robot hits it. In other words the information about the environment has a local character, i.e., information only about the immediate surrounding is available. Since the information about the environment is incomplete, the plan cannot be pre-planned, and so its global optimality is ruled out. The advantage of using a point robot in two dimensions is the reduction of options available to it when encountering an obstacle, since it can turn left or right along the obstacle boundary. The only available input information includes the robot‘s own co-ordinates and those of the target. The environment is a plane with a set of obstacles, finite in number, that do not touch one another, and the points start *S* and target *T* in it. The motion capabilities of the mobile robot include the following: (i) move from starting point to target point on a straight line, (ii) move along the obstacle boundary, and (iii) stop. A local direction is either left or right once an obstacle has been hit. The algorithm defines a hit point *H* on an obstacle, while moving along a straight line towards the target, and the robot contacts the obstacle at the point *H*. It defines a leave point *L* on an obstacle, while it leaves the obstacle at the point *L* in order to continue its straight line path toward *T*. All hit and leave points are recorded during the exploration of the environment from *S* to *T*. Two algorithms are proposed, *BUG1* and *BUG2*, the former being rather conservative and latter more humanlike.

Iyengar etal [Iyengar (1987)] incorporated learning by moving the robot through an unknown environment and incrementally constructing a visibility graph of the environment. Learning takes place as the robot visits a number of destination points. The point robot can memorise the vertices of the obstacles it visits. This technique assumes that the unexplored terrain is filled with disjoint convex polygonal obstacles which are mutually nonintersecting and no touching. In this approach, the visibility graph $VG$ is constructed as follows: (i) *V* is the set of vertices of the obstacles; (ii) *E* is the set of edges of the graph. A line connecting the vertices $V_i$ and $V_j$ forms an edge $(V_i, V_j) \in E$. Initially, *VG* is completely unknown. During the navigation process the robot constructs the learned visibility graph (*LVG*) consisting of the vertices and edges that have been traversed. Eventually the *LVG* should converge to *VG*. Suppose that the robot has to move from a straight point *S* to a target point *T*. Initially the robot moves along the straight line *ST* designed by the unit vector $\hat{n}_{st}$ until it meets the nearest obstacle.

Subsequently, it circumnavigates the obstacle using a local navigation strategy. Upon meeting the obstacle two directions are possible in going round it, $\hat{n}_1$ and $\hat{n}_2$. The local optimisation criterion is defined as follows: *max* $(\hat{n}_{st} . \hat{n})$, where $\hat{n}$ is unit vector along the direction of motion. Meanwhile, the procedure also incorporates a learning phase to acquire *VG*. This method was extended by Kant and Zucker [Kant (1988)] and Kyriakopoulos and Saridis [Kyriakopoulos and Saridis (1993)].

An approach based on vector field histogram was developed by Borenstein and Koren [Borenstein and Koren (1991)]. This approach generates both a direction and speed for the robot. However the disadvantage is that this approach ignores velocity and acceleration constraints on robot's motion.

Recently behaviour based approach has been developed. Yen and Pfluger [Yen and Pfluger (1995)] reflects the increasing interest in the potential of fuzzy logic in mobile robot navigation. The mobile robot considered by the authors in their work has a behavioural repertoire of actions, each suitable for a particular situation and is essentially a reactive control approach to selection of behaviours. There are two directions of movement to choose from, the desired direction and the allowable direction. The decision making process is to take the fuzzy variable that results as the minimum at any point between the desired direction and allowable direction, and then take the maximum point in that variable. The defuzzification process may produce a direction of movement that is close to the desired one, when it could have avoided collision by not altering its direction at all and thus save energy. Situation like this arises because fuzzy controls are used without the need to optimise some measure of performance and the lack of a detailed mathematical model may result in manoeuvres beyond the capabilities of the mobile robot.

Research in this area by Michon and Denis [Michon (2001)] provides insights into how landmarks are used for human navigation and what are considered to be key route points.

Another alternative strategy is that developed by Kuipers which combined qualitative and quantitative space representations into the spatial semantic hierarchy (SSH), to build a representation model of large-scale space. [Kuipers2000] The SSH models the human cognitive map and serves as a method for robot exploration and map building.

Kuipers and Byun [Kuipers (1991)] implemented the control, topological and metrical levels on a simulated robot with 16 range-sensors similar to the sensory system of the robot used in this work. In their experiments, the robot explores a simulated office environment and identifies a set of locally distinctive topological map elements (20 places and 23 edges), which are uses to construct a complete topological map.

Dai and Lawton used landmarks for qualitative robot navigation. [Dai (1993)] However, the algorithms completely ignore range estimates of landmarks. Determining robot position, when landmarks are indistinct, depends on recognizing the distribution of landmarks surrounding a robot.

Levitt and Lawton took a similar approach with the intention to prevent the error in robot navigation from accumulating, as it is common in techniques such as triangulation, ranging sensors, stereo techniques, dead reckoning, inertial navigation, correspondence of map data with the robot's locations and local obstacle avoidance techniques. [Levitt (1990)] Inspired by human and animal navigation performances which use extremely poor range estimates and very coarse angular information, Levitt and Lawton used distinctive landmarks to navigate a robot. A major constraint in spatial representation is the uncertainty in the absolute position of almost everything in the world, including robot position and the position of landmarks.

Nourbakhsh et al. used color vision and odometric information for robot navigation [Nourbakhsh (1999)]. To alleviate the poor information content of range-finding sensors, they made use of artificial colored landmarks that the robot is able to recognize. This is another landmark-based technique that works well if modifications to the environment are allowed. Suitable environments would be museums (such as the one used for the robot, Sage of the Carnegie Museum of Natural History, which was one of the applications of Nourbakhsh's technique), office buildings, etc. [Nourbakhsh (1999)].

A solution to the problem of creating an accurate global topological map comes from Simhon and Dudek. They chose to use quantitative environment information to create local metric maps, which they used to form a global map [Simhon (1998)]. Assuming that a method to generate the local map already exists, the problem becomes one of choosing an area suitable to create a local map which is accurate and effective. Hence, they derived a measure of distinctiveness between different regions in the environment.

A qualitative approach to robot navigation using schematic maps is that of Freksa et. al. [Freksa (2000)]. Freksa et. al. formulate a model that uses qualitative spatial relations for robot navigation based on schematic map. A schematic map is a reduced detail topographical map as opposed to a sketch map, which is generated from abstract mental concepts and verbal descriptions.

In [Muller (2000)] an approach is proposed that enables a wheelchair to follow a route in a building, based on landmark recognition. Muller et. al. assume that the robot operates strictly in a building environment, so the problem of landmark recognition becomes one of corridor detection and orientation. The landmarks used in the [Muller (2000)] approach are: wall in front, corridor left, corridor right, door left and door right. The route is composed of a series of statements, each of which defines a navigation command (turn right, follow corridor, enter left door, etc.) upon sensing a particular arrangement of landmarks (corridor right, right hand bend, etc.). The algorithms of this work attempt to generalize the definition of landmarks to be any object or combination of objects and not strictly corridors, walls or entrances. Sonar sensors are also used as the sole source of sensory input. The algorithm is further developed by Chronis. Chronis' work [Chronis (2002, 2007)] provides a different perspective for path planning. As mentioned in Chapter 1, the navigation issues always high computational calculation and involve with high level sensors such as vision, etc. Chronis uses low level sensors to generate linguistic descriptions of spatial relations between robot and environment. His eventual goal is to use these linguistic descriptions for navigation of the mobile robot in an unstructured, unknown, and possibly dynamic environment. Thus an exact model of the environment is not built, but an approximation of the local environment is generated.

## 2.5 The Proposed Path Planning in This Thesis

The approach, proposed in this thesis, which can be applied in a cluttered, unknown environment, provides the mobile robot with an egocentric structured environment, such as left wall, front wall and right end corner etc. In Chapter 3, a method is developed to obtain a representation of the structure, and its classification with the help of rules. The robot should recognize its structure in terms of egocentric spatial relations between itself and obstruction in its environment. It is also proposed with fault tolerance ability, i.e. determination of classification of structures can be carried out when sonar sensor fails. To achieve these, 3 types of status for the sonar sensors are defined: a) sonar sensor is said to be engaged when the reading is less than the maximum (object/obstacle detected), b) Sonar sensor is in operational status but has maximum distance return, this situation means the sonar sensor detects no obstructions, c) Sonar sensor fails, in the situation sonar sensor have no return value. The details of detection and classification algorithms will be introduced in Chapter 3.

However, uncertainty in the detection can be overcome by using the notion of consensus of sensors while at the same time increasing the tolerance to sensor failures. The other objective of the algorithm is to provide a fault-tolerant operation of the robot. This thesis presents a navigation method which overcomes uncertainty in the detection of objects and surfaces by using consensus sensors, while at the same time increasing the tolerance to sensor failures. This technique can be used for navigation in both known and unknown environments. It is developed to ensure that the robot moves from its current location to some desired locations. It should be noted that the primary aim is to perform the task but not to increase the computational overhead.

For mobile robot to successfully plan its path in unstructured or unknown environments it not only needs to gather the right quality and quantity of information, also needs to make sense of this information, and make use of this information efficiently in order to be able to modify its path. Our approach fuses the information from eight sonar sensors in a consensus method and thus provides geometric information to make sense of the environment. The advantages of homogeneous sensors fusion are reducing fusion complexities, and allowing fusion at the sensor data level. Furthermore, the multi configuration sensors can be applied in the construction of geometric based world representations. [Bajcsy & Allen (1985), Chen & Medioni (1991), Porrill (1988)]

The path planning scheme developed in this thesis can be applied in an unknown environment. The nominal path is a straight line from the starting point which is the mobile robot's initial position to the desired position. Our approach has enhanced detection ability when the mobile robot meets a clutter of obstacles in an unknown environment; it gathers the geometric information of the surrounding obstacles and classifies these obstacles into structures such as wall, corner and etc. The eight sonar sensors can be thought as a network of consensus sensors which make a decision and at the same time they can be thought as eight individual sonar sensors which are mounted on the robot. During the observation, the decision will be made by at least three votes from 3 different sensors. The approach allows a cluttered unknown environment to be transformed into basic structured environment in mobile robot's view. For example, in the situation of Figure 2.1, when the mobile robot meets the obstacles, its decision making process, using information from sonar sensors, will view obstructions as a whole blocking like a front wall. The mobile robot classifies those obstacles as walls (virtual walls). The avoidance strategy will lead the robot to avoid the clutter of obstacles as avoiding a wall. Once the mobile robot reaches a clear area, the mobile robot will head to the goal point again. However the obstructions are not a front wall, so it can be thought as a virtual wall. When in the same situation, Chronis' approach detects the situation in stage 1 Figure 2.1

as three objects.

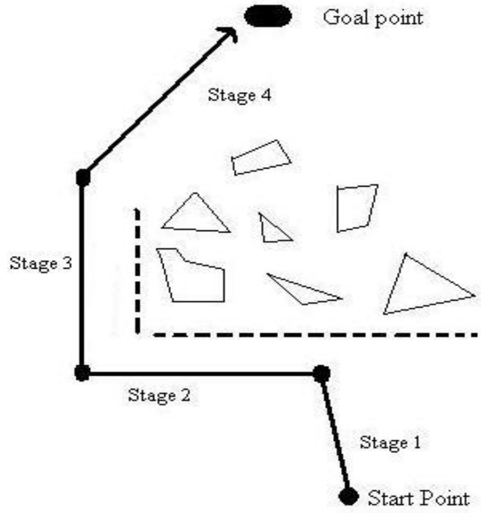


Figure 2.1 illustrates the proposed path planing scheme.

# Chapter 3 Rules for Classification into Structures by Using Multiple Homogenous Sonar Sensors

## 3.1 Introduction

In this Chapter, Saphira environment and Pioneer simulator are introduced. The Pioneer simulator has eight homogenous sonar sensors which can be thought as distributed and networked sensors. The essential geometric and distance information for designing the classification rules can be gathered by fusing homogenous sonar sensors. The details can be found in later sections in this chapter. The rules classify the environment into a simpler world which can be rebuilt with 16 primitive structures: an open area, an object, a virtual object, a front wall, a virtual front wall, a left wall, a virtual left wall, a right wall, a virtual right wall, a corridor, a virtual corridor, a left corner, a virtual left corner, a right corner, a virtual right corner and a dead-end.

To navigate in the environment, a mobile robot must sense the environment in order to know where it is. An autonomous mobile robot has to be able to sense the environment, and make decisions with no human interference. In order to make a correct decision, it is important that the robot senses its environment accurately, e.g. by using sonar and vision sensors. Using sonar sensor, it can map the seafloor [Morgan (1998)]. Even for an object in space, its position, size, shape, velocity and direction of motion can be determined. Kuc and Siegel have shown sonar scans also can use in everyday indoor environments [Kuc and Siegel (1987)]. In this thesis, the mobile robot uses sonar sensors for detecting obstructions.

Mobile robot uses fused information to construct a structure view of the environment. The fusion combines individual distance and geometric information, which are aggregated under certain rules by each sonar sensor. Most of the research in the area of multi-sensor robotics focuses on the application of multiple homogeneous sensors. The homogeneous sensors fusion is computationally efficient in that the fusion complexities are reduced, and it allows fusion at the sensor data level (low level, a common information representation). Furthermore, the multi-configuration sensors were suggested in the construction of geometry-based world representations, which include multiple ultrasound sensors and multiple vision sensors. Single-sensor systems are inherently limited in their ability to provide a consistently reliable stream of information due to view-point occlusions, innate uncertainty, limited operating range and accuracy, individual sensor biases, spurious errors, and other factors attributable to employing only one sensory modality. In contrast, a properly employed synergistic multi-sensor configuration possesses the potential to provide:

(i) information from multiple viewpoints.

(ii) robust and fault-tolerant operation.

(iii) extended spatial and temporal coverage.

(iv) diverse properties of the environment.

(v) improved accuracy.

(vi) less ambiguity.

In this thesis, Sahpira system is used for development and verification. Saphira is a robotics application developed at the Pioneer mobile robot platform. The Pioneer robot has eight sonar sensors and we can consider these eight sensors as distributed and networked sensors. The algorithm for fusing information requires voting from each sensor and then reaching an agreement. In this thesis, the rules are designed for developing the geometric and distance information of these eight sensors. The problem of distributed consensus has historically existed in many diverse areas: Communication Networks [Mehyar, Spanos, Pongsajapan, Low and Murray (2005)], [Liu and Yang (2003)], Control Theory [Jadbabaie, Lin, and Morse (2003)], and Parallel Computation [Tsitsiklis (1984)], [Bertsekas, and Tsitsiklis (1989)].

### 3.2 Saphira Environment and the Pioneer Simulator

Saphira has a C-like language, known as Colbert, for writing robot control programs. With Colbert, users can quickly write and debug complex control procedures, which are called Activities. Activities have a finite-state semantics that makes them particularly suited to representing procedural knowledge of sequences of actions. Activities can start and stop directing robot actions and other activities. Activities are coordinated by the Colbert executive, which supports concurrent processing of activities. Colbert comes with a runtime evaluation environment in which users can interactively view their programs, edit and rerun them.

Saphira comes with a software simulator of the physical robot and its environment, which allows users to debug applications conveniently. The simulator has realistic error models for the sonar sensors and wheel encoders (Table 3.1). The reality is such that, if a client program works with the simulator, it will work on the physical robot as well. The disadvantage of the simulator is that the environment model is an abstraction of the real world, with simple 2-D linear segments in place of the complex geometrical objects that real robot will encounter in the real world.

| Parameter | Pioneer Value | Description |
|---|---|---|
| **EncodeJitter** | 0.01 | Error in distance |
| **AngleJitter** | 0.02 | Error in angular position |

Table 3.1 Saphire build-in error model.

#### 3.2.1 Representation of Space

Mobile robots operate in a geometric space, and the representation of that space is critical to robot’s performance. There are two main geometrical representations: Local representation and Global representation. The Local Space is an egocentric coordinate system a few meters in radius centered on the robot. The Global Space representation provides a larger perspective where objects are represented as a part of the robot's environment. The local space gives the robot a sense of its immediate surroundings. The test environment can be constructed in 2-D models, which is known as world models in Saphira. A world description file is a plain text document typically stored with the file name suffix *.wld*, which describes the size and contents of a simulated world. World models are abstractions of the real world, with linear segments representing the vertical surfaces of corridors, hallways, and the objects in them. The simulation environment 2 used in this thesis is shown in Figure 3.1 (the corresponding .wld file in the appendix):

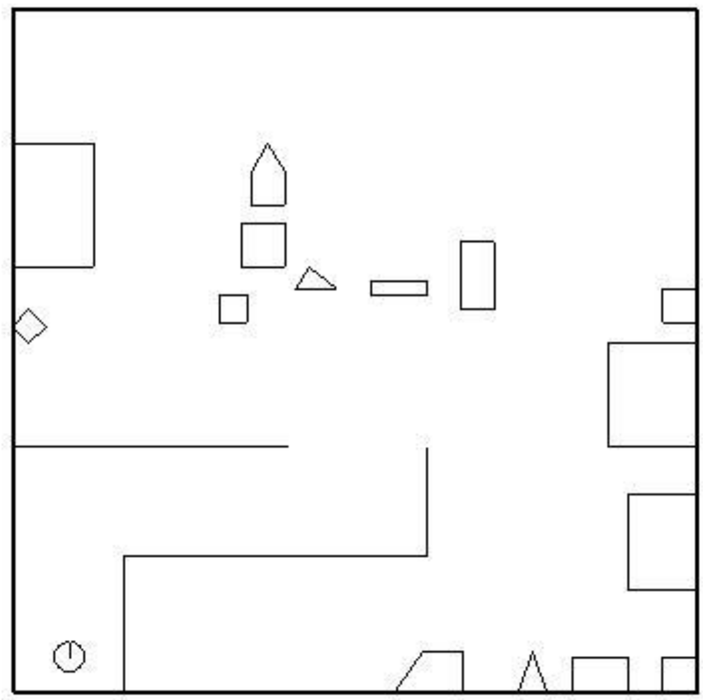

Figure 3.1 The simulation environment 2.

### 3.2.2 Sonar Sensors on the Robot

Pioneer robots are a family of mobile robots with either two-wheel or four-wheel-drive. They are all small, intelligent robots, whose hardware architecture was originally developed by Kurt Konolige, of SRI International. In this thesis the simulator uses the Pioneer 2 parameters. The diameter of the Pioneer 2 is 250 mm. The Pioneer 2 mobile robot has 8 sonar sensors in the front, distributed at a total of 180 degrees; range data obtained from these 8 sonar sensors can provide basic geometric and distance information about the environment. One motivation for using sonar sensors for mobile robot navigation comes from the impressive ultrasonic sensing capabilities of bats, which rely on echolocation to determine their position and to hunt their prey [Armingol, 2004]. The essential condition for geometric navigation of a mobile robot is the ability to determine its position. The robot obtains the input data from the sonar sensors by movement in the map or virtual/real environment. The sonar sensor systems generally calculate distance by using the time of flight (TOF) method. Define *d*, as the distance between the sonar senor and object, *c* as the speed of sound and *t* as the time of sound wave traveling on both ways, the two way distance can be calculated by speed of sound multiply by the time traveling; the single way distance can be calculated refer Figure 3.2.

$$d = \frac{1}{2}ct \tag{3.1}$$

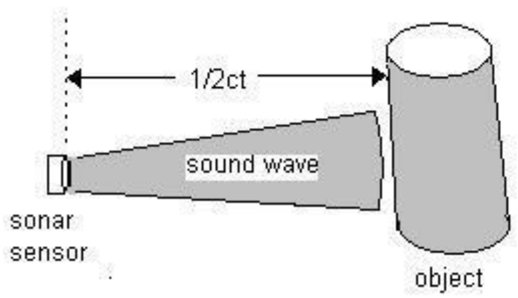


Figure 3.2 Sonar distance- measuring concept

In the simulation environment, a *.p* file, as shown in table 3.2, is used to store the basic parameters of the robot, and it must be loaded to form a simulated pioneer class robot. This file contains both geometric and kinematic details for the simulator such as radius of the robot and speeding parameter, etc. Figure 3.3 shows the Pioneer 2 robot in the simulation.

```
;; Parameters for the Pioneer 2 DX Mobile Robot
AngleConvFactor 0.001534    ; radians per angular unit (2PI/4096)
DistConvFactor    0.826         ; mm returned by P2
VelConvFactor      1.0          ; mm/sec returned by P2
RobotRadius         250.0         ; radius in mm
RobotDiagonal      120.0         ; half-height to diagonal of octagon
Holonomic              1                ; turns in own radius
MaxRVelocity       500.0         ; degrees per second
MaxVelocity         2200.0        ; mm per second
RangeConvFactor 0.268           ; sonar range returned in mm
;; Robot class, subclass
Class      Pioneer
Subclass p2dx
SonarNum 8 ; 8 total sonars
;; These are for the eight front sonars: six front, two sides
;; Sonar parameters
;;           SonarNum N is number of sonars
;;           SonarUnit I X Y TH is unit I (0 to N-1) description
;;                          X, Y are position of sonar in mm, TH is bearing in degrees
;;                #     x     y     th
;;-------------------------
SonarUnit    0    115    130    90
SonarUnit    1    155    115    50
SonarUnit    2    190    80    30
SonarUnit    3    210    25    10
SonarUnit    4    210    -25    -10
SonarUnit    5    190    -80    -30
SonarUnit    6    155    -115    -50
SonarUnit    7    115    -130    -90
FrontBuffer 20
SideBuffer    40
```

Table 3.2 An example of a *.p* file.

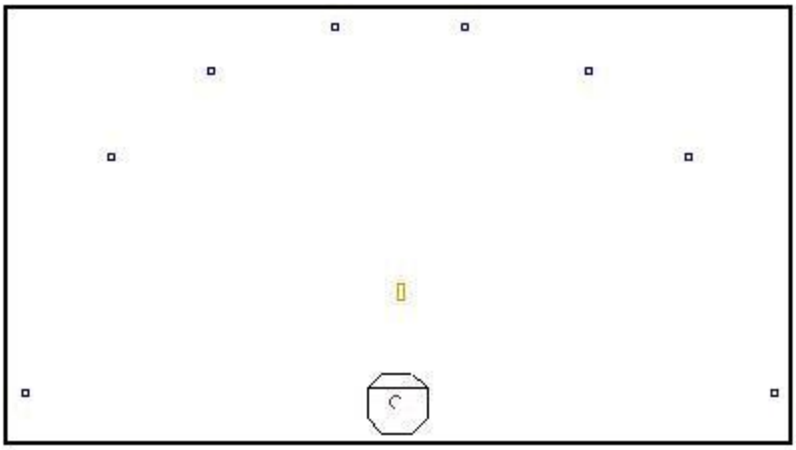

Figure 3.3 The simulated pioneer 2 robot.

## 3.3 Geometric Information of Multi-sonar-sensor Configuration

A single sonar sensor only provides distance to the object without any other information, such as shape or orientation. In this thesis the robot considered has eight sonar sensors mounted on it and numbered from 0 to 7. Readings are defined as SR0, SR1 …SR7. From now on, these sensors will be labeled as Si, i=0…7. The robot can be thought of as a local coordinating system. The centre of the mobile robot will be the origin of the coordinating system. A sonar detected point can be transformed as a point with coordinates $(x_i, y_i)$ in the local system:

$$x_i = (SRi + Radius) * \cos\theta \tag{3.2}$$

$$y_i = (SRi + Radius) * \sin\theta \tag{3.3}$$

Note: *Radius* is the mobile robot's radius and $\theta$ is the angle the sonar sensor mounted.

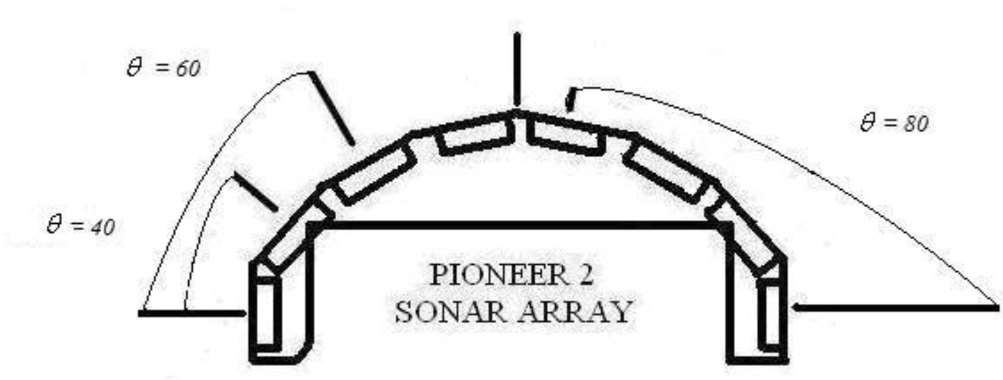


Figure 3.4 Illuminates the $\theta$ value.

$\theta$ has always a positive value, e.g $\theta$ = 40 when the angle being considered is between S0 and S1; the robot is symmetrical, the angle between S6 and S7 is 40 degree. The angle between S0 and S2 is 60 degree, and the angle between S5 and S7 is 60 degree, hence $\theta$ = 60. The angle between S0 and S3 and the angle between S7 and S4 are 80 degree, hence $\theta$ = 80, see Figure 3.4.

$X_i$ and $Y_i$ are the notations for the distance from the detected point to the centre of robot; and the absolute value of $x_i$ and $y_i$. The distances from the detected point to the robot are shown in the following formula:

$$\begin{aligned} xSRi &= |(SRi + Radius) * \cos\theta - Radius| \\ &= |x_i - Radius| \end{aligned} \qquad (3.4)$$

And

$$\begin{aligned} ySRi &= |(SRi + Radius) * \sin\theta - Radius| \\ &= |y_i - Radius| \end{aligned} \qquad (3.5)$$

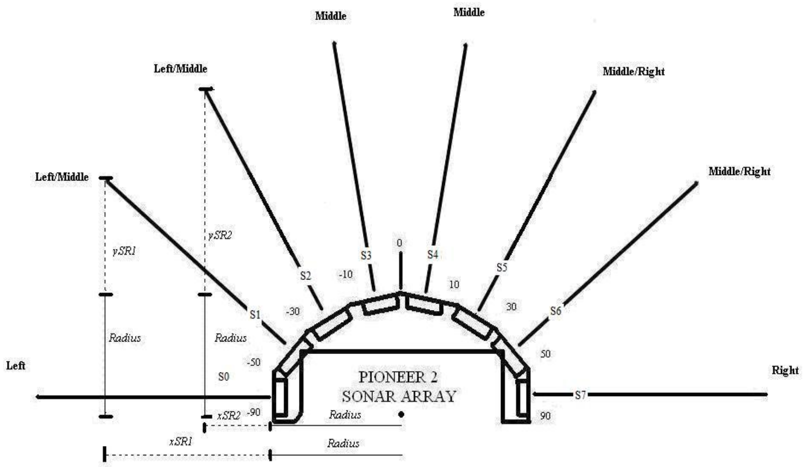


Figure 3.5 Classification of sensors and number of sensors (S0 to S7)

The sonar sensor can detect an obstacle from a distance of up to 2999 mm away. Consider that S1 detects an obstacle and returns a distance reading of 2999 mm. According to equations 3.4 and 3.5, $xSR_1 = 2240\ mm$ and $ySR_1 = 1840\ mm$, (see Figure 3.5). If the returned reading is 2999mm then the value of S6 is the same as S1, so $xSR_6 = 2240\ mm$ and $ySR_6 = 1840\ mm$. So the remaining sensor value as follow: $xSR_2 = 1375\ mm$ and $ySR_2 = 2565\ mm$ and $xSR_5 = 1375\ mm$ and $ySR_5 = 2565\ mm$; $xSR_3 = 315\ mm$ and $ySR_3 = 2950\ mm$ and $xSR_4 = 315\ mm$ and $ySR_4 = 2950\ mm$. These distance information is complementary for positioning how close the object is to the robot horizontally and vertically.

In case when the mobile robot is not too close to the obstruction, a safety parameter of distance, e.g. 200 *mm*, can be set, see Figure 3.6. Details of a safety module will be introduced in Chapter 6. Consider the case of each sonar sensor's reading when $xSR_i$ and $ySR_i = 200\ mm$; the vaules of $SR_i$ can be calculated by substituting $xSR_i$ and $ySR_i = 200\ mm$ into equation 3.4 and 3.5; For example, when $xSR_1$ and $xSR_6 = 200\ mm$, then $SR_1$ and $SR_6 = 338\ mm$, this distance information which provided by S1 or S6 detection range 338 *mm* is a boundary value, if SR1 or SR6 is less than *338 mm*, there is a possible horizontally collision to the robot. Thus we calculate these boundary values for each sensor: when $xSR_2$ and $xSR_5 = 200\ mm$, the boundary value of $SR_2$ and $SR_5 = 650\ mm$;

when $xSR_3$ and $xSR_4 = 200\ mm$, the boundary value of $SR_3$ and $SR_4 = 2340\ mm$. On the other side, when $ySR_1$ and $ySR_6 = 200\ mm$, the boundary value of $SR_1$ and $SR_6 = 450\ mm$, and there indicates possible vertical collision to the robot. These boundary values for each sensor can be calculated: when $ySR_2$ and $ySR_5 = 200\ mm$, the boundary value of $SR_2$ and $SR_5 = 270\ mm$; when $ySR_3$ and $ySR_4 = 200\ mm$, the boundary value of $SR_3$ and $SR_4 = 207\ mm$.

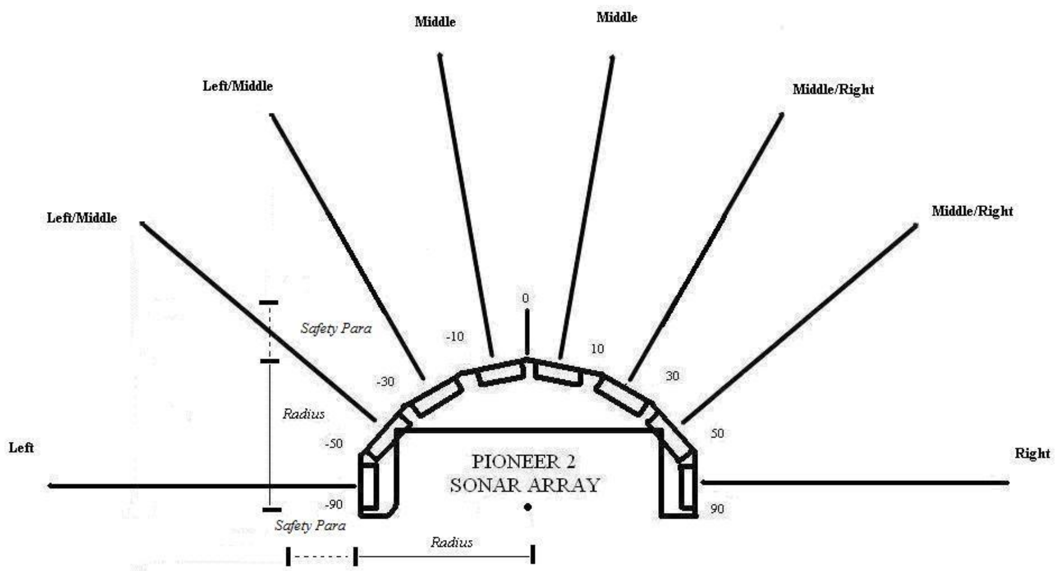


Figure 3.6 illuminates the safety parameter.

When $SR_3$ and $SR_4$ = 2340 mm, $xSR_3$ and $xSR_4$ = 200 mm, it is a boundary value, and only one sonar sensor 3 or 4 detects object. When $xSR_3$ or $xSR_4$ > 200 mm, the conclusion is that the obstruction will not block the front direction during the travel; when xSR3 or $xSR_4$ < 200 mm, it means that the obstruction will block the front direction during the travel. S3 and S4 are critical for the front detection. S0 and S7 are critical for left and right directions. The rest of the sonar sensors will provide complementary information in further consensus management.

### 3.4 Rules for Detection of Objects

The robot is equipped with eight sonar sensors. During the detection, the information from the sensors is processed in order to make decisions. The sonar sensor information is fused with geometric information in order to classify the environment into a basic structured view. The conventional approach such as Huygen's principle uses single sonar sensor scans to verify the predictions of their model. One key conclusion from their work is that corners and walls produce responses that cannot be distinguished from a single sensing location. John Leonar [J.Leonard (1991)] extends this idea by using rotated sonar scans to differentiate the target types. He put forward three types of target: planes, corners, and edges. Lindsay Kleeman and Roman Kuc [Kleeman and Kuc (1995)] found the minimum requirements of an array of transducers established to identify primitive reflector types in an indoor environment. The reflector types considered in their approach are planes, corners and edges. The corner is assumed to be a concave right angle intersection of two planes. An edge represents physical objects such as convex corners and high curvature surfaces. Their finding by using two independent transmitters and two independent receivers are sufficient to discriminate planes, corners and edges in two dimensions [Kleeman and Kuc (1995)]. In this thesis, the sonar sensors are classified three directions: left detection sensors, middle detection sensors and right detection sensors (see Figure 3.5). S0 mainly detects obstructions in the left of the robot, and it is the crucial sensor for left detection, which can detect the obstruction up to 2999mm to the very left of robot (The sensor number is shown in Figure 3.6). S1 and S2 can provide complementary information to detect a front wall or left hand corner. S3 and S4 are usually for front detection. S7 detects right side of the robot. S7 is the crucial sensor for right detection, which can detect the obstruction up to 2999mm to the very right of robot. S5 and S6 can provide complementary information to detect a front wall or right hand corner.

In the environment, whether sonar sensor is engaged or not determines the presence of an object. Therefore, once the reading of a sonar sensor is shorter than the maximum reading (open range status), it means that the mobile robot detects obstacles or the presence of a possible structure (The sonar sensor is engaged and in detection status). If any sensor is engaged, it can represent obstruction in the direction. The consensus status of other sonar sensors then can be gathered to distinguish a different structure Thus the following rules 1-3:

**Rule 1:** ***If*** *(readings of eight sonar sensors == maximum value)*

***Then (****open area)*

**Rule 2:** ***If*** *(only the readings of one sonar sensor < maximum value)*

***Then*** *(single obstacle)*

**Rule 3:** ***If*** *(only the readings of two continuous sonar sensors < maximum value,)*

***Then*** *(large single obstacle)*

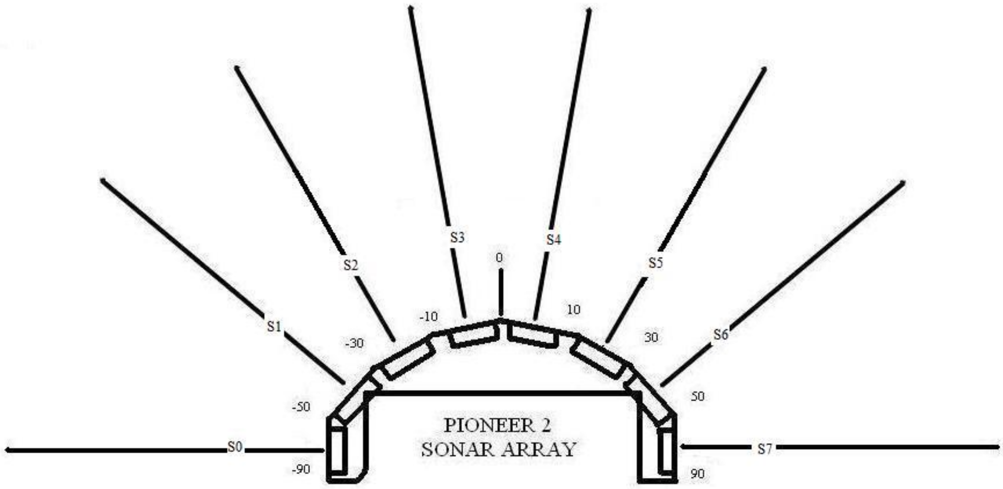


Figure 3.7, The mobile robot in open area (Rule 1).

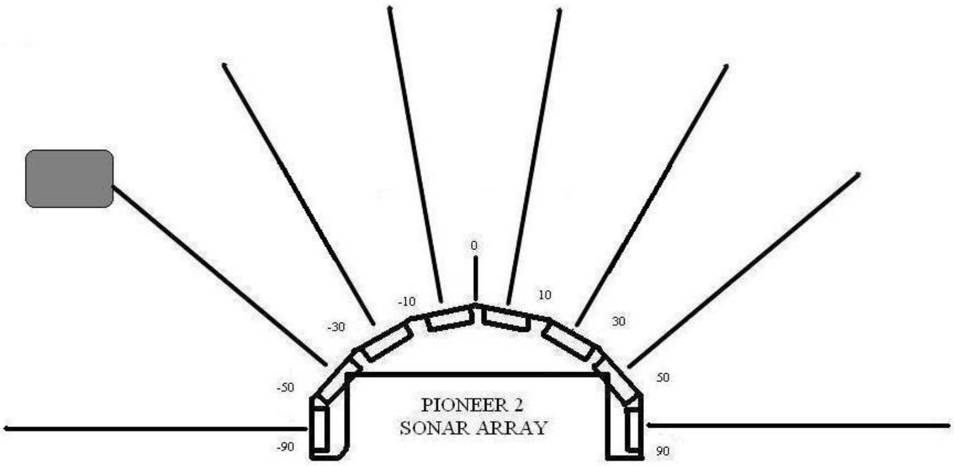


Figure 3.8 A single object detected by the mobile robot (Rule 2)

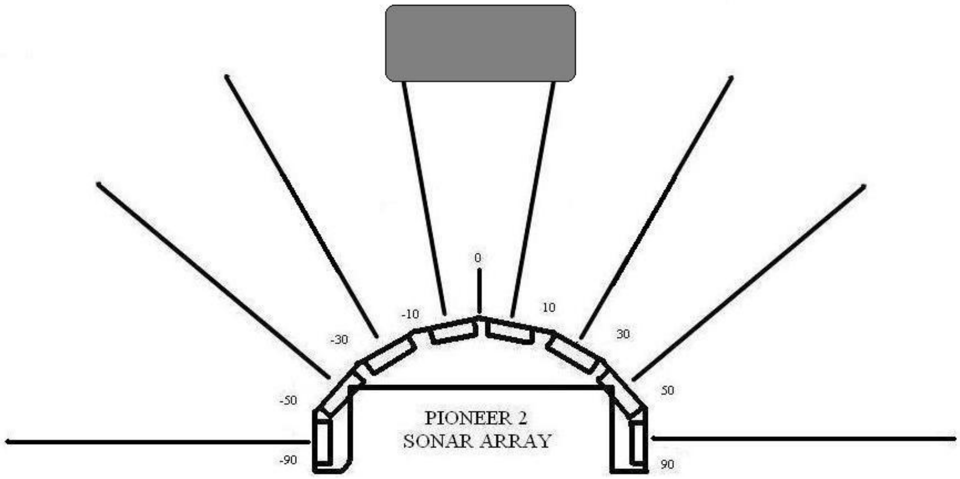


Figure 3.9 A bigger obstacle detected by the mobile robot (Rule 3)

The obstacle detected could be in different shapes and forms. However, in some special circumstances when the sensor reading satisfies the obstacle detection rules, the situation will be classified as robot detecting an obstacle. In an extreme situation, large obstacles detected by three continuous sonar sensors may be considered as surface or part of a large structure. On the other hand, multiple obstacles may be classified as parts of structure as well. The primitive can be defined as surface or wall structure in this thesis.

### 3.5 Wall on the Sides

**Rule 4:** ***If*** *(the only readings of sonar sensor 0,1,2< maximum value)*

***Then*** *(left wall)*

**Rule 5:** ***If*** *(the only readings of sonar sensor 7,6, 5 < maximum value,)*

***Then*** *(right wall)*

Consider a mobile robot travelling along a wall, sensors S0, S1 and S2 are all engaged at the same time. This confirms that there is an obstruction located to the left of the robot. From the egocentric view of the mobile robot, as long as all the left sensors have detected something, it will consider the obstacle to be a wall-like structure.

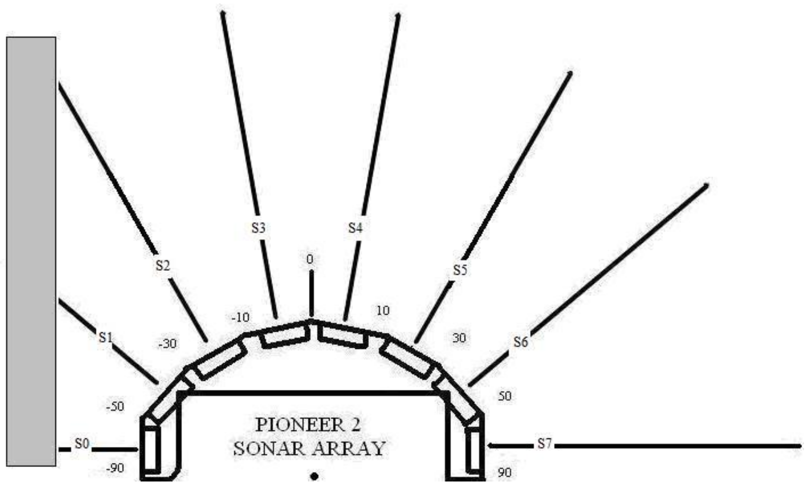


Figure 3.10 Detection of left surface (Rule 4).

In the same situation the mobile robot detects a right wall, while the robot travels and S5, S6 and S7 are all engaged at the same time. This confirms there is obstruction located to the right of the robot. According to the rules, it confirms a wall located in the right side of the mobile robot. Of course, there is always a situation when there are obstacles and each is sensed by one sensor, while the special case will be discussed in Chapter 4.

### 3.6 Obstruction in Front – a Wall Ahead

Figures 3.11 and 3.12 present different situations of mobile robot meeting obstruction in the front. The situation of Figure 3.11 shows the sufficient condition for the mobile robot to confirm a front wall. That is, up to six sonar sensors are engaged to confirm a front wall obstruction. The situation of Figure 3.12 shows the necessary condition of the mobile robot to confirm a front wall, which takes at least three sonar sensors to confirm a surface.

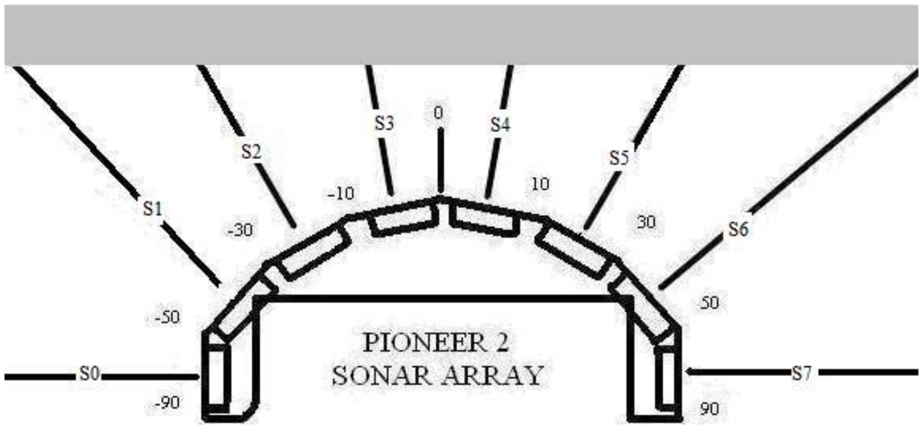


Figure 3.11 Detection for a front wall with engagement of all possible sensors (Rule 6).

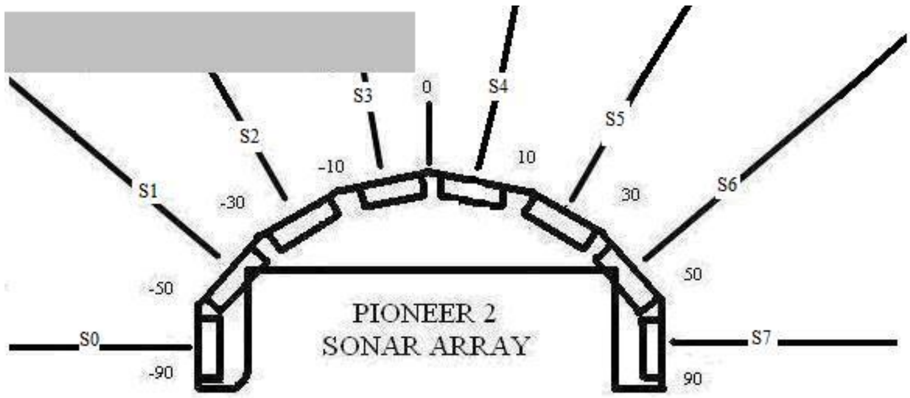


Figure 3.12 Detection of a front wall.

| |
|---|
| **Rule 6:** ***If*** ***(***(*(the sonar sensor 0, 7) == maximum value) && (readings of three adjacent sonar sensors< maximum value))*<br>***Then*** *(front wall)* |

The algorithm for determining a front surface: S3 or S4 are not compulsorily engaged at the same time, but at least one available; S0 and S7 are complementarily required as returning the maximum detection range.

### 3.7 Detection of Corner, Corridor and Dead-end

When the mobile robot travels towards a corner as the case shown in Figure 3.13, the obstruction can be thought as being made up of a side wall and a front wall. The dash in Figure 3.13 shows the other situation when the front obstruction can block a wider area.

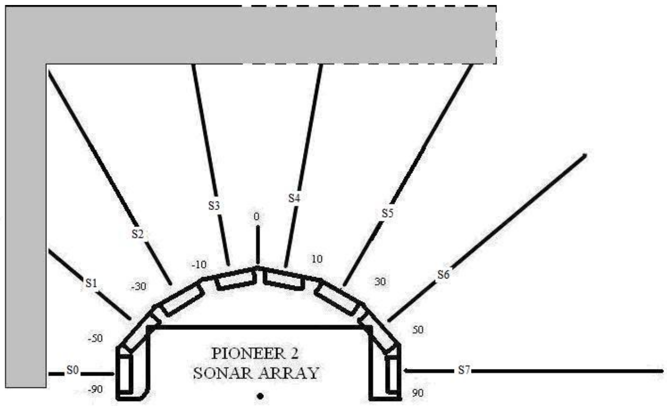


Figure 3.13 A left-hand corner detected by the mobile robot (Rule 7)

| |
|---|
| **Rule 7:** ***If (****(the readings of sonar sensor 0,1,2,3, (4, (5, (6))) < maximum valu**e)*** |
| ***Then*** *(left corner)* |
| **Rule 8:** ***If (****(the readings of sonar sensor 7,6,5,4,(3, (2, (1))) < maximum value)* |
| ***Then*** *(right corner.)* |

In the case of detecting a corner for example, detecting a left hand corner, it requires at least S0, S1, S2, S3 are engaged and S7 must be in the open range status, however, in addition if S4, S5 and S6 are sequentially engaged which means the *route is gradually blocked by the corner*. The same concept applies to detecting a right hand corner, which requires at least S7, S6, S5 and S4 are engaged and S0 must be in the open range status, however, if S1, S2 and S3 are sequentially engaged, this situation indicates the route is *gradually* blocked by the corner. In the situation where all sensors are blocked and the mobile robot detects a *dead-end* (see to Figure 3.15.) The dead end can be thought as two sides walls intersect with a front wall. According to the rules the structure of the corridor will be thought as two parallel walls, (see to Figure 3.14).

**Rule 9:** ***If** (((the readings of sonar sensor 0, 1, 2, 5, 6, 7) < maximum value) && ((the readings of sonar sensor 3,4)< maximum value))*

***Then** (corridor.)*

**Rule 10:*****If** (the readings of eight sonar sensor < maximum value)*

***Then** (dead-end.)*

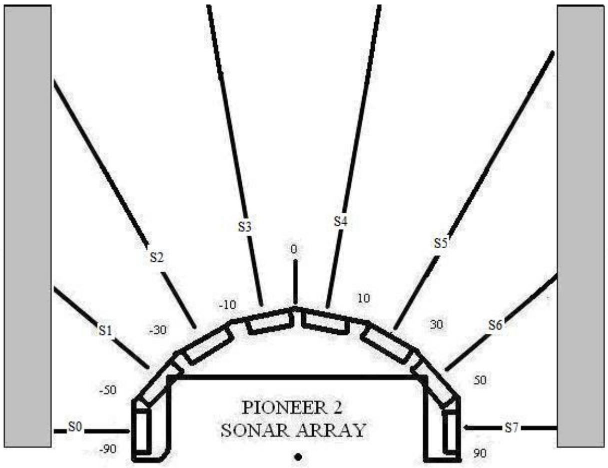


Figure 3.14 The mobile robot travelling in a corridor (Rule 9).

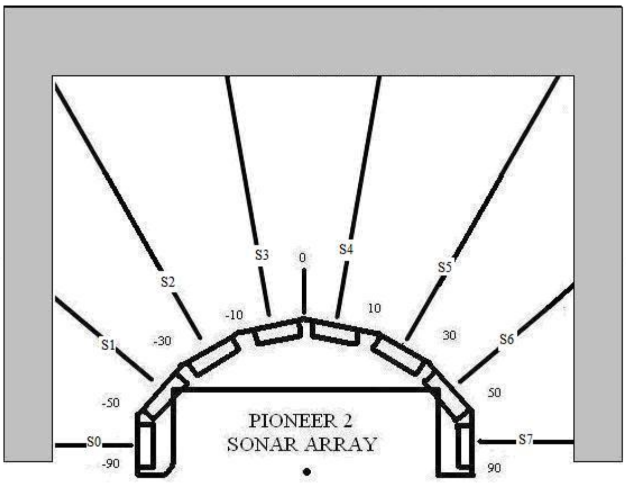


Figure 3.15 A dead-end detected by the mobile robot (Rule 10)

### 3.8 Conclusion

The multiple sensor information reduces ambiguity and increases tolerance to sensor failures in operation. The rules enable the robot to construct structures of environment and navigate in an environment by using low level sensors. It detects the structures instantly. And it requires little computation and very small amount of memory. In constructing of the structures, a decision is made on the basis of fusing information of each individual sensor. In the structured environment, two engaged adjacent sensors can confirm a larger object; and three engaged adjacent sensors can confirm a surface. A surface is a primitive in this thesis, the other structure such as corridor, corner, etc can be made by this primitive. In chapter 4, this approach is applied to a cluttered environment, and it is shown the appropriate decisions to be made. In Chapter 7, this approach is also compared with Chronis' approach.

# Chapter 4 Robustness of the Proposed Rules for Detection Structures in Cluttered Environment

## 4.1 Introduction

Navigation in a cluttered environment, the traditional Planning-Sensing-Modelling-Acting approaches are not effective. Instead, local navigation strategies that tightly couple the sensor information with control actions must be used for the robot to successfully achieve its mission [Sgorbissa & Arkin (2003)]. The control complexity is overcome by decomposing the navigation control problem into simpler and better-defined sub-problems (behaviour) that can be controlled independently and in parallel [Arkin (1998)]. This has attracted the interests of many roboticsts and has been used for industrial process control [Linder (1998)]. It is important that algorithms for navigation control in cluttered environments require not to computational overhead and not to cause sluggish response. As discussed in Chapter 3, the information is gathered from eight sensors and fused by using rules so as to make sense of the local environment.

To fuse information correctly depends on the sensors' providing correct information. However there are situations when the information from the sensors is incorrect. The proposed rules incorporate methods which can overcome sensor failures and sonar sensor uncertainty. This implies that the strategy has an element of robustness and tolerance to faults. The system has a component monitoring readings of the sonar sensors. When the reading is zero, it fails to provide distance information; the fault-tolerant module treats the respective sonar sensor as a faulty sensor.

## 4.2 Structure Classification in a Cluttered Environment

Figure 4.1 shows a situation where the mobile robot is approaching a group of cluttered objects. The sensors S2, S3, S4 and S5 are engaged. The algorithm classifies the possible situations as a front wall (see Section 3.4). The algorithm relies on the status of the sonar sensors, and the distance information is ignored at this stage. There are three possible statuses of the sensor: (a) open range status which means the sonar sensor return as the maximum distance value; (b) detected status which means the sonar distance value between 1-2999; and (c) failure status which means the reading of the sonar sensor is constant 0. Though obstacles shown in Figure 4.1 are randomly located, the layout of the obstacles is the key to make the algorithm to classify the situation as a front wall.

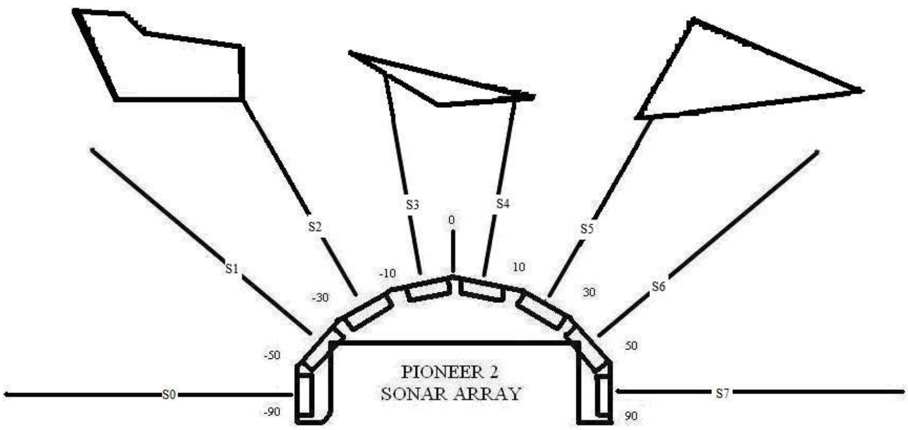


Figure 4.1 Cluttered objects classified as a front wall by the mobile robot

Consider the situation shown in Figure 4.2, S2, S3, and S5 are engaged, and S4 shows the open range. According to the classification rules, the situation is that the robot is moving towards two dis-neighbouring obstacles. S2 and S3 are in detected status, and the two objects detected by S2 and S3 can be regarded as one bigger obstacle. S5 is also in a detected status, and the object detected by S5 can be taken as small single obstacle.

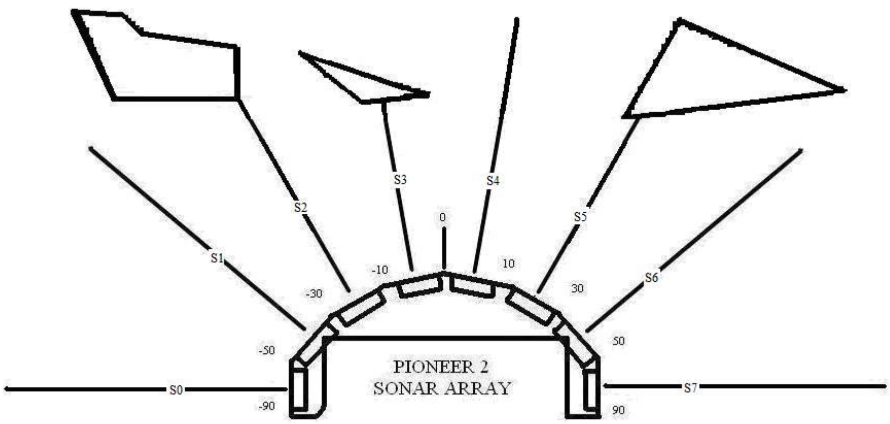


Figure 4.2 Approaching wall- like dis-neighbouring objects

In reality, the situation can be thought alternatively as a front wall with a small gap. When the sonar reading reaches the maximum range, the sonar beam covers a cone area of detecting direction, and the maximum width of the cone can be taken as the maximum width of the gap. It can be calculated when the sonar sensor reaches the maximum reading. The single sonar sensor detected space is not enough for the mobile robot to pass through safely. Even if there are two gaps, space is enough for robot passing through. Thus the algorithm still classifies the single gap situation as a front wall though it is called a *virtual* front wall.

It should be noted that in this situation, a virtual front wall is detected by three sonar sensors but not neighbouring as required by the rules (see Chapter 3). *Thus a similar situation might happen if S4 provides wrong information or has failed.* This situation can be distinguished from the normal front wall detection; *it is detected by three sonar sensors but not neighbouring as the previous rules required.* It is a front obstruction as the S0 and S7 are returning the maximum value and at least one of crucial front detection sensor 3 or 4 is engaged. In Chapter 6, the local path planning scheme controls the robot to go around a group of cluttered obstacles.

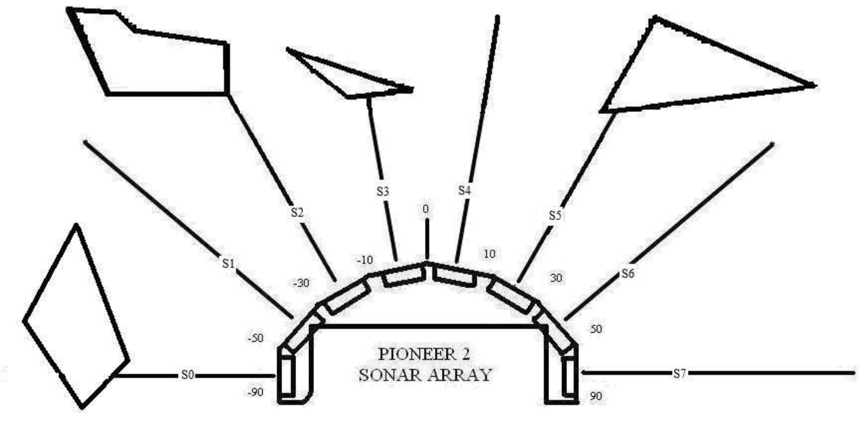


Figure 4.3 Approaching corner-like cluttered objects

The situation presents in Figure 4.3 where S0, S2, S3, and S5 are engaged, and the obstacles are in a corner-like layout. S0, S1, S2 in engaged status and the sonar bounced form a left wall, and the sonar from S2, S3, S4 and S5 bounced form a front wall. There are two crucial detection sensors in the detection status and they are the most important votes for classifying the left end corner structure. Apply the same principle to the above classification of a virtual front wall. S0 and S2 are in the detection status and can confirm the left wall with a gap; S2, S3 and S5 are in the detection status and can confirm the front wall with a gap; therefore the virtual corner can be classified as a left end corner with gaps. In the local path planning phase, gaps are ignored; the mobile robot will go around the virtual corner rather than crossing it. The same concept can be applied to the detection of other virtual structures, such as a virtual corridor or a virtual dead-end.

### 4.3 Structure Classification with Partial Sonar Sensor Failures

In the situation of bad sensor readings and sensor failures, it is imperative that the robot carries on functioning albeit with reduced efficiency. The algorithm is extended to deal with this problem via a scheme developed to increase the tolerance to sensor failures and enable the robot to carry on navigating. The faults involved in this project are those which occurred when the sonar sensor has a zero reading. When a sonar sensor fails, its reading is zero.

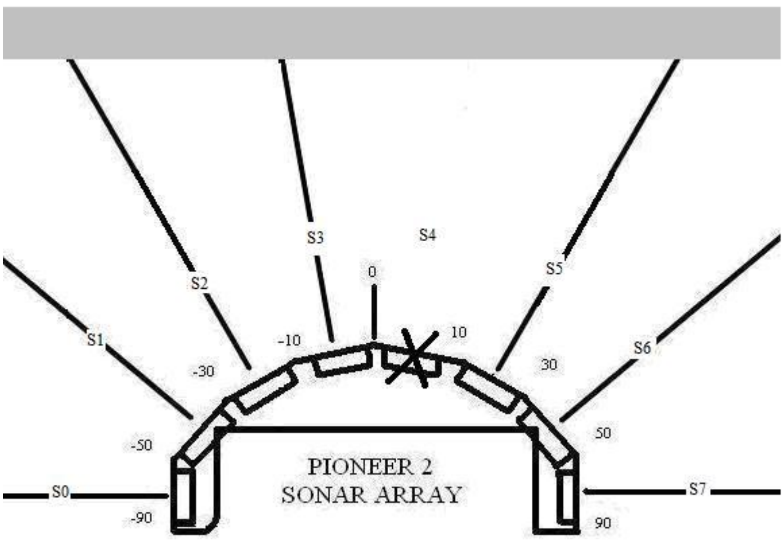


Figure 4.4 Approaching a plane with sonar sensor 4 failed

Consider the situation in Figure 4.4. The mobile robot is approaching the wall and sonar sensor 4 has failed. The mobile robot cannot determine whether there is obstruction in the detection direction. The algorithm will assume the sensor 4 to be still in the detection status since S2, S3, S4 and S5 are all engaged in this case. According to the rules mentioned in the former chapter, the algorithm classifies the situation as a virtual front wall. And the sonar sensor 4 fails to provide any information.

### 4.4 Detection in Cluttered Environment with Partial Sensor Failure

It is a more complex case when the robot has partial sensors failures in a cluttered environment. The robot will construct the virtual structures with comparatively less information gathered from the environment.

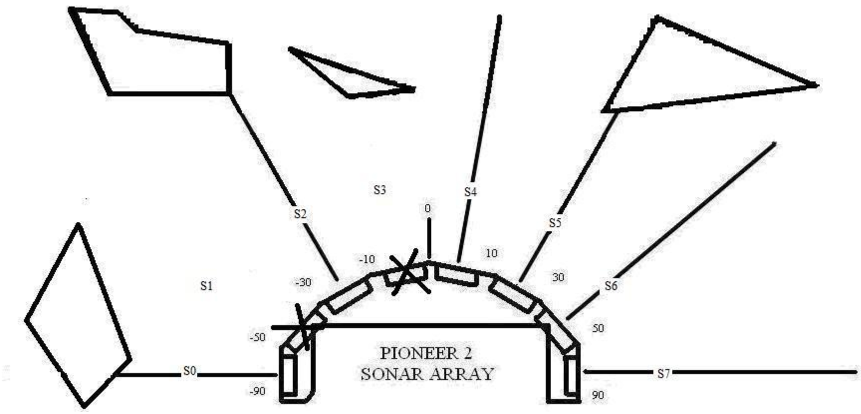


Figure 4.5 Approaching a cluttered environment with the failure of Sonar Sensor 2

The situation in Figure 4.5 shows that both S1 and S3 have failed. Thus, the mobile robot only knows S0, and S2 and S5 are engaged. This could mean there are three objects in the environment (see Chronis' approach in Chapter 7). It is difficult for the mobile robot to construct a virtual structure in the situation as this. The rules transfer the failure status of S1 and S3 into the detection status. The situation of Figure 4.5 is transferred into the situation of figure 4.6 (It is not concern sonar reading at this stage only transfer sensor status). The dotted triangle and rectangle are the assumptive obstacles detected by S1 and S3. The dotted lines are the assumed sonar sensor readings. The situation of Figure 4.6 is classified as a virtual corner by the algorithm. In the later path planning phase, the predefined virtual corner avoidance strategy will drive the robot go around it.

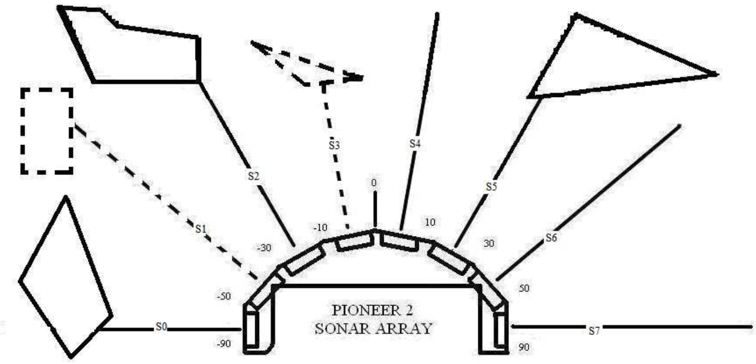


Figure 4.6 Sonar Sensors 1 and 3 being assumed to be in working order

### 4.5 Conclusion

This chapter discusses the situation when the robot is in a cluttered environment and some of the sonar sensors fail. Two non-adjacent engaged sensors can form a virtual surface in condition that the two must be adjacent with a failed sensor in the middle of these two non-adjacent sensors. The rules can allow up to three failed sensors at one time with restriction to the locations of the failed sensors. In the course of detection, the crucial sensors S0 and S7 cannot fail at any time. Otherwise, the robot will lose the main guidance of left or right direction. The sonar sensors 3 and 4 must not fail at the same time. Otherwise, it may result in crucial damage to the front. The algorithm allows no more than three sonar sensor failures at one time, thus the general rules are:

- The sensors 0 and 7 cannot fail at any time.
- The sensors 3 and 4 cannot fail at the same time.
- No two adjacent sonar sensors can fail at the same time.

Two adjacent sonar sensors' failing at the same time can cause large blind space and may cause collision. The worst case for a robot is with failed sensor in a cluttered unknown environment, because it gathers less information from the environment. The methods will be experimented in a specific situation in Chapter 5 and in a larger global context in Chapter 7.

# Chapter 5 Experiments of Building Structures

## 5.1 Introduction

In this chapter and the next, rules are developed to enable a robot to move from the current location to some desired locations. The scheme uses the techniques developed in Chapters 3 and 4 to make sense of its environment (Chapter 5) and navigate around obstacles (Chapter 6). The navigation algorithm applied in an unknown environment does not attempt to optimise the length of the path because safety issue is more important than a minimum length path and computational overhead needs to be kept to a minimum (Ghatee 2009; Stachniss 2003). The scheme has two stages: (a) global path planning and (b) local obstacle avoidance. Global path planning requires a pre-planned nominal path and a set of nominal points generated by the planning algorithm to schedule the robot's actions (see Washington 1999; Tompkins 2005). In this thesis the nominal path is always a straight line from the origin to the desired location (Chapter 1). All deviations are measured from this straight line. The scheme constructs local instantaneous structures by using the geometric information of the obstacles along the nominal path. In addition, the scheme enhances fault-tolerant ability in overcoming partial sensors failures.

## 5.2 The Proposed Approach

The techniques introduced in the earlier chapters transform an unknown cluttered environment into a structured environment. The techniques isolate the cluttered obstructions, and re-generate a path which goes around the cluttered obstructions. The example of rules can be seen from Figure 1.3. The scheme classifies obstacles into 16 primitive structures (from Chapters 3 and 4): an open area, an object, a virtual object, a front wall, a virtual front wall, a left wall, a virtual left wall, a right wall, a virtual right wall, a corridor, a virtual corridor, a left corner, a virtual left corner, a right corner, a virtual right corner and a dead-end. The scheme is capable of tolerating at most three sonar sensors' failures at the same time. The systemic design shows in Figure 5.1.

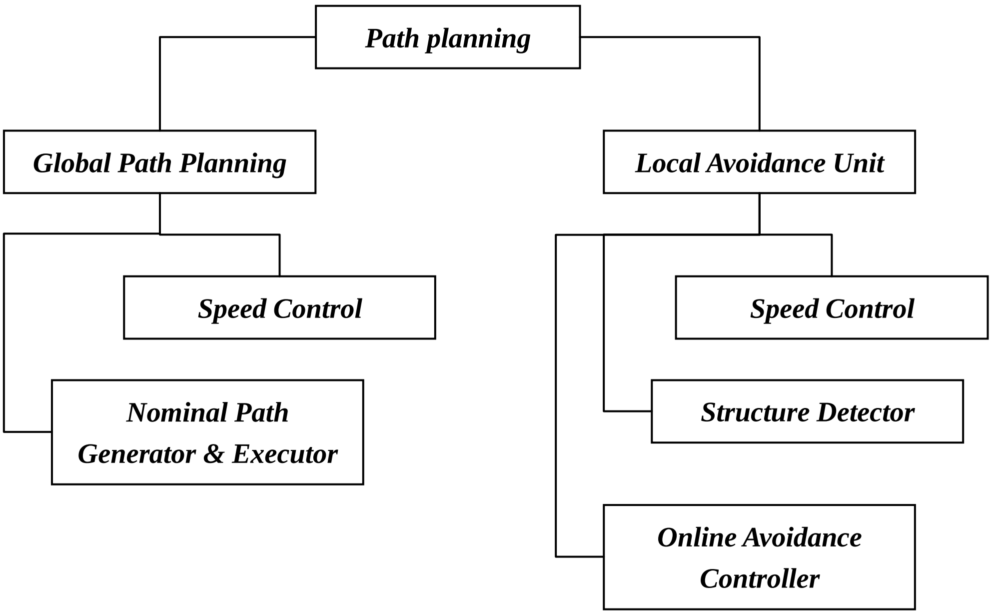


Figure 5.1 The systemic design of the proposed scheme

The features of the scheme are:

- *It allows for online path planning, which is designed for structured and cluttered unknown environments.*
- *It constructs the structures base on the notions of consensus sensors fusion along with the use of geometric information regarding the configuration of the robot and the layout of sensors.*
- *It is robust and can tolerate the failure of up to three sensors during the travel.*

## 5.3 Global Path Planning

The scheme is designed for a mobile robot travelling in an unknown environment. The global path planning unit generates a straight line from the starting point to the goal point.

The global path planning unit sets the mobile robot's current coordinates as starting point (the coordinates can get by input *sfRobot.ax* and *sfRobot.ay* commands in Saphira system). The goal point will be assigned by the system or a human. The goal point's coordinates will be relative to the mobile robot's coordinate system. In Figure 5.2, $(X, Y)$ presents the location's coordinates.

$$Distance\ travel = \sqrt{x^2 + y^2} \quad (5.1)$$

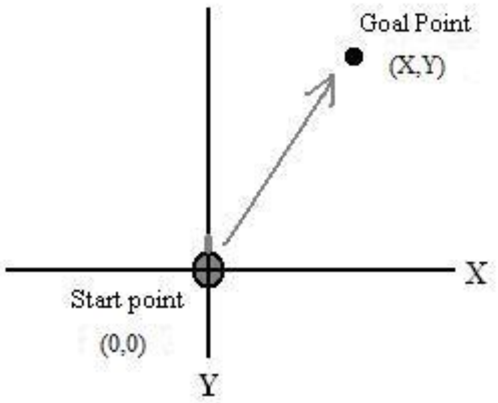


Figure 5.2 Global coordinate system

## 5.4 Local Path Planning

The local path planning unit is composed of three components: (a) the speed control unit —controlling the speed when the mobile robot meets the obstruction; (b) structure detection unit —the main unit which classifies the obstructions into basic structures; and (c) online avoidance unit — the control for the mobile robot to avoid collision and detected structures. The online avoidance unit stores individual avoidance strategies for each type of structure. The scheme assigns individual strategies to produce avoidance procedures (see Chapter 6).

### 5.4.1 The Structure Detection Unit of Building Local Structures

The structure detection unit is based on the techniques developed in Chapters 3 and 4, which gathers the surrounding geometric information of the mobile robot and fuses with the information of each sensor to classify cluttered objects into basic structures.

### 5.4.2 Detection of Object

When the mobile robot is in an open area and all the sonar sensors return the maximum reading, the mobile robot is free to travel and there is no possibility of collision. The scheme takes the open area as a basic structure. The condition for an open area is that all the sensors return the maximum reading (see Figure 5.3).

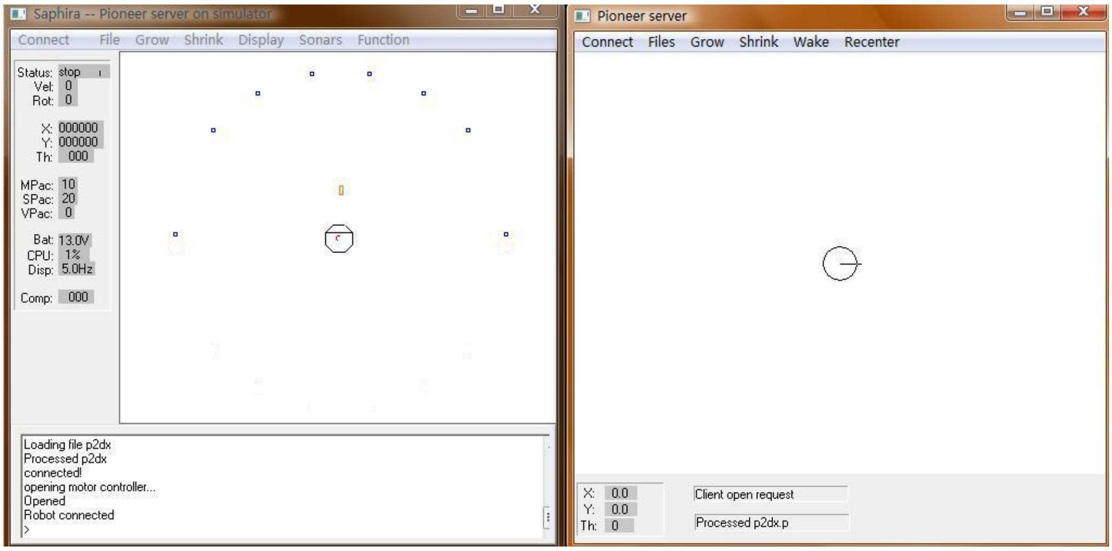


Figure 5.3 The mobile robot in an open area

In Figure 5.3 the dots in the left window show the sonar detection points and they are all showing the maximum readings, indicating the robot in an open area. In the left box of Figure 5.3 the rectangle above the robot indicates the heading direction.

There are two situations of obstacles which are defined in this thesis — a small obstacle and a large obstacle. The obstacle detected by a single sonar sensor is called a small obstacle. A large obstacle is classified when there are two or more neighbouring sonar sensors are engaged. They do not mean “small” and “large” in size. They are differentiated by the number of using detected sonar sensors (see Figure 5.4).

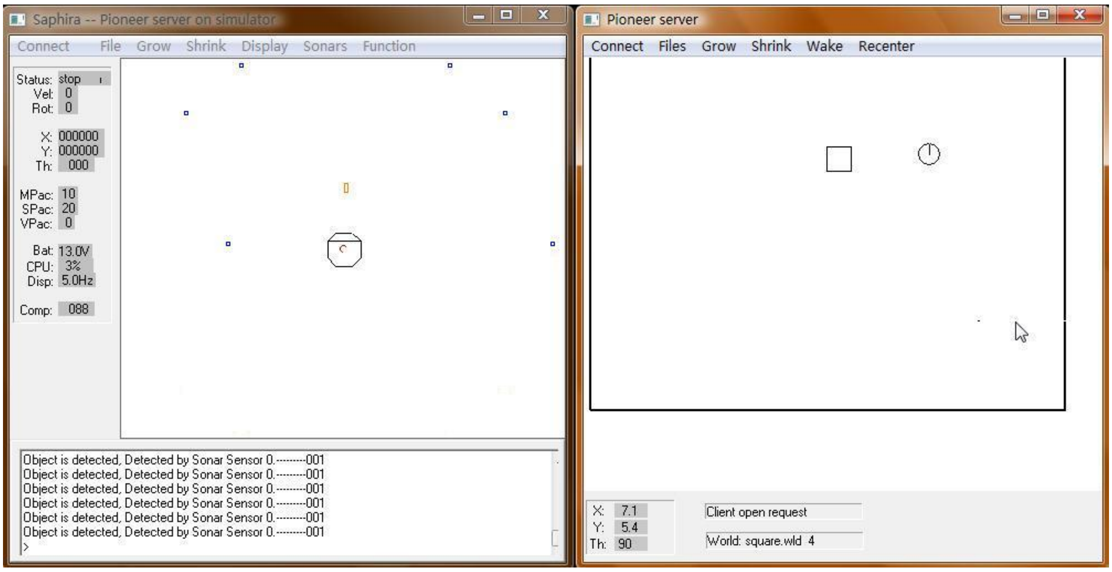


Figure 5.4 Detection of an object by a single sonar sensor.

Figure 5.4 shows the situation when an obstacle is located around the robot, only S0 detects the obstacle with the range information and shows the detecting status in the left window.

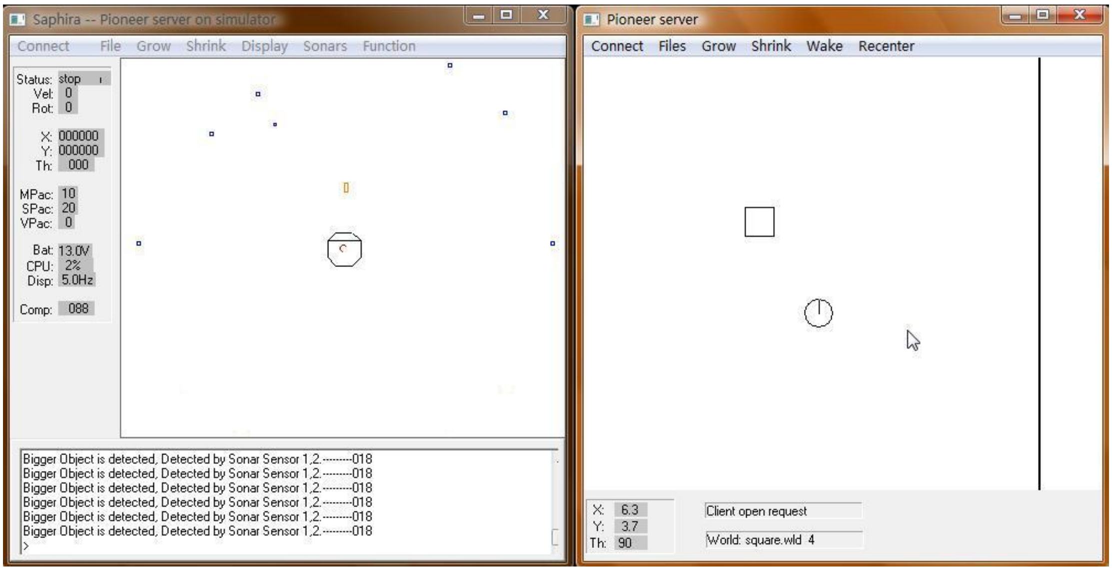


Figure 5.5 Detection of an object by two neighbouring sonar sensors.

The small obstacle is detected by only one sensor, and the maximum detection range of sonar sensor is 2999 mm. The maximum detection width is the sound wave width when the reading reaches the maximum range. The sound wave width is fixed when the type and model of sonar sensors are the same. It can be calculated by the equation below.

$$Sound\ Wave\ Width = 2*(SR_i + 250)/\tan 80^o; \qquad (5.2)$$

The sound wave width approximately equals 1130 mm. The widest obstacle that can be detected by a single sonar sensor then will be no more than 1130 mm. In the case of a large obstacle being detected, the maximum width of obstacle can be calculated to be approximately 2220 mm. There are special situations since the angle between S0 and S1 is 40 degree, when S0 and S1 detect the large obstacle, it can be thought as a left wall (vice versa to the right side). So the rules require at least two sensors for detecting a side surface.

As discussed in Chapter 1, in the case of multiple sonar returns, there is a question of whether the adjacent sonar readings are from a single obstacle or multiple obstacles. In the stage of the obstacle detection, two adjacent sensors engaged are thought to come from one large obstacle. The obstacle can be defined under conditions below:

- One or two neighbouring sonar sensors are engaged.
- Other sonar sensors are in the open range.

Whatever the real structure or shape of the obstacle is, we only consider the detected as an obstacle. In this situation, where the robot is in the open area and there is one sonar sensor failure, the system will show that there is a small obstacle located in the direction of the failure sensor points. It also can be detected as a large obstacle when the adjacent sonar sensor is engaged. Refer to Figures 5.6 and 5.7.

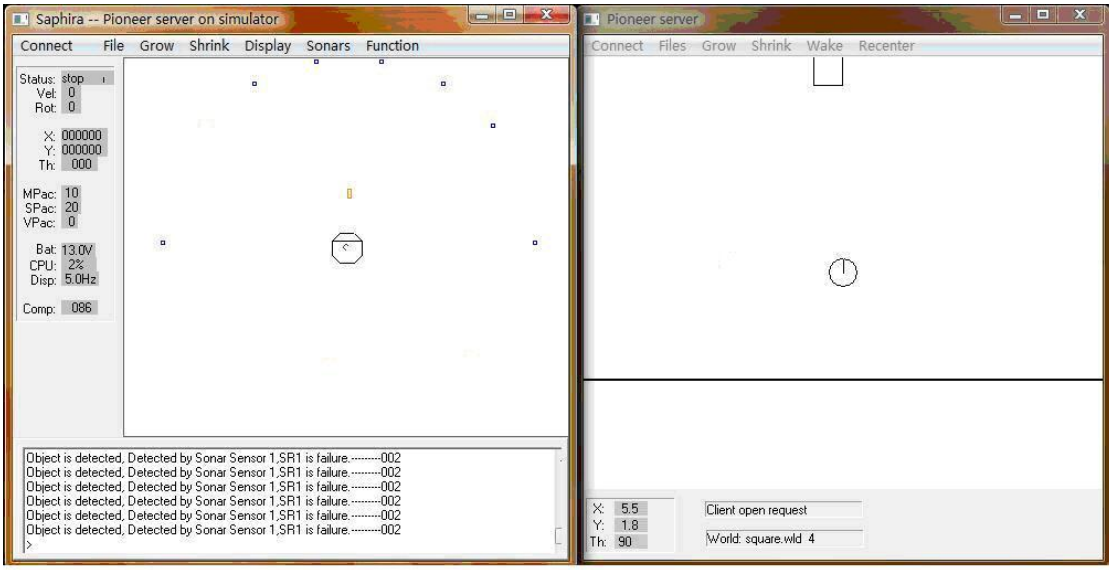


Figure 5.6 Detected an object with S1 failed.

The robot cannot notice its existence since S1 fails, and the robot takes the situation as having detected an obstacle located in the direction of S1. In the left window of Figure 5.6, the test program indicates: *Objects are detected; Detected by Sonar Sensor 1; and SR1 failure.* Figure 5.7 shows when the robot closes to the object, S2 detects the object and S1 fails. The rules assume S1 in detection status. Thus the situation is identified by the robot as a larger obstacle. In the left window of Figure 5.7, the test program shows: *A large Object is detected; Detected by Sonar Sensor 1, 2. Sensor fails*.

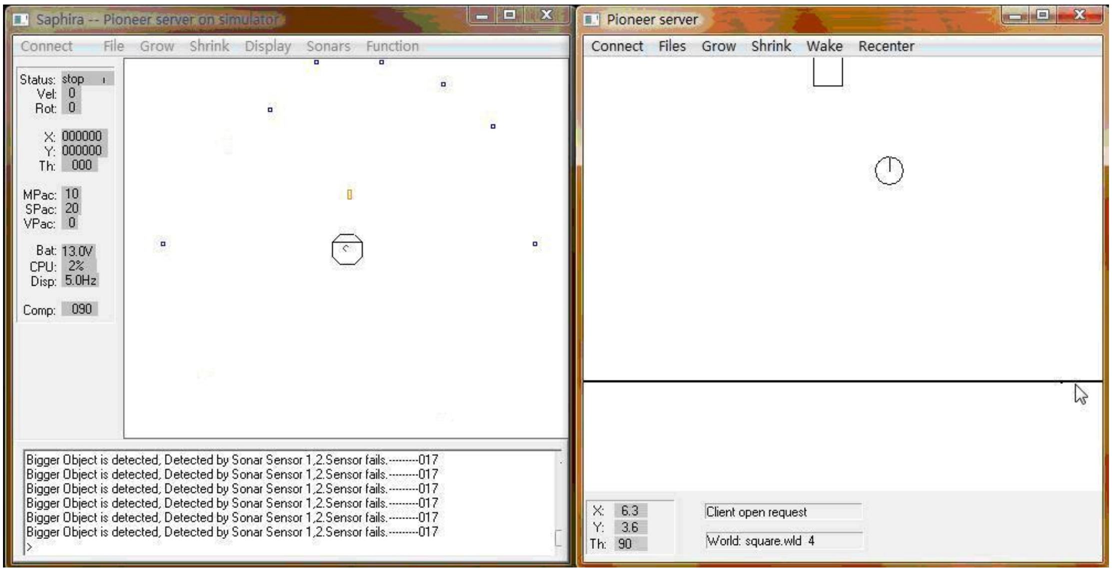


Figure 5.7 Object detected in the situation of two dis-neighbouring sonar sensors failed.

### 5.4.3 Detection of Surface

The rules suggest that at least two neighbouring sonar sensors are engaged in order to form a surface. When all detection sensors are in working order, the sufficient condition to form a wall is more than two neighbouring sensors engaged at the same time. The rules allow up to three sonar sensors to fail at the same time.

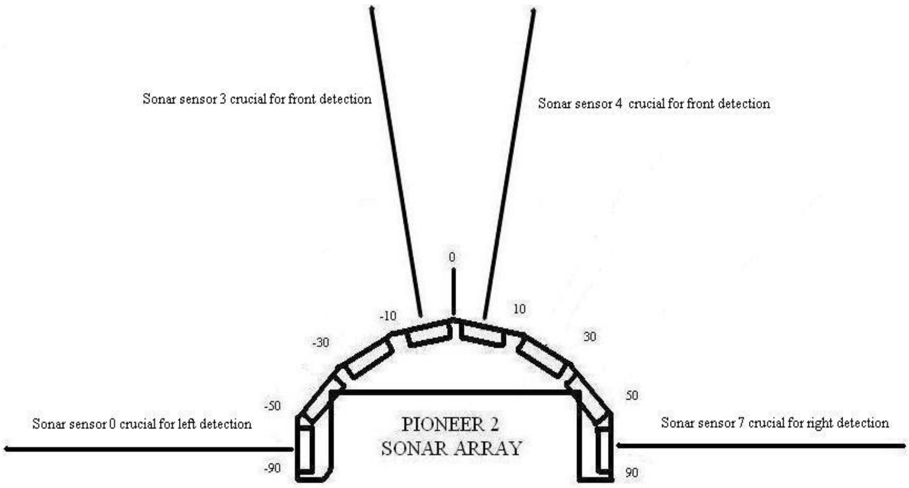


Figure 5.8 The four crucial sonar sensors.

As mentioned in Chapter 3 the sonar sensors in three directions: left detection sensors, middle detection sensors and right detection sensors (see Figure 3.5). The reading from S0 and S7 mainly detect obstructions in the left and right of the robot; and they are crucial sensors for left and right detection, which can detect the obstruction up to 2999mm on the very left and right of the robot (see Figure 5.8). S0 and S7 are the most important votes for left and right structure indication. The S1 and S2 can provide complementary information for detection of front and left obstruction. The S5 and S6 can provide complementary information for detection of front and right obstruction (see Figure 3.5 and Equations 3.4 and 3.5).

When S1, S2, S5, and S6 reach 2999 mm, these distances can be calculated as follows: $xSR_1=2240\ mm$ and $ySR_1=1840\ mm$, $xSR_2=1375\ mm$ and $ySR_2=2564\ mm$, $xSR_5=1375\ m$

and $ySR_5=2564$ *mm*, $xSR_6=2240$ *mm* and $ySR_6=1840$ *mm*. The information as such can provide knowledge of how far the obstruction is from the front and the sides. However, they do not provide any directional information. S3 and S4 usually work for the front detection, and which are of vital importance (see Figure 5.8). When S3 and S4 reach the maximum range, $xSR_3=314$ *mm* and $ySR_3=2950$ *mm* and $xSR_4=314$ *mm* and $ySR_4=2950$ *mm*. The distance information providing for the sides can be ignored. During the detection the sensors S0 and S7 can never fail. Otherwise, it will cause the robot to have no left or right action abilities. S3 and S4 must not fail at the same time, as discussed in Chapter 4, The algorithm allows no more than three sonar sensors at the same time:

- The crucial sensor S0 and S7 cannot fail at any time.
- The crucial sensor S3 and S4 cannot fail at the same time.
- Two adjacent sonar sensors cannot fail at the same time.

The failure of two neighboring sonar sensors at the same time can cause loss of detection over a large area and also result in loss of accuracy.

#### 5.4.3.1 Detection of a Front Wall

A surface as a simplest structure and the sufficient condition for detection of a surface is that there are at least three sensors engaged. When the front wall is very close to the robot, up to six sensors are engaged, i.e. S1... S6, see Figure 5.9. In the left window of Figure 5.9 the test program shows: *Front wall is detected by sonar sensor 1, 2…6*. The angle between S1 and S6 are fixed which is $100^o$. Thus the minimum detection width (MDi represents minimum detection width of i sensors, i <= 6.) can be calculated by the formula:

$$MD6 = 2*(ySR3 +250)/tan40o; \qquad (5.3)$$

$$MD6 = 2*(ySR4 +250)/tan40o; \qquad (5.4)$$

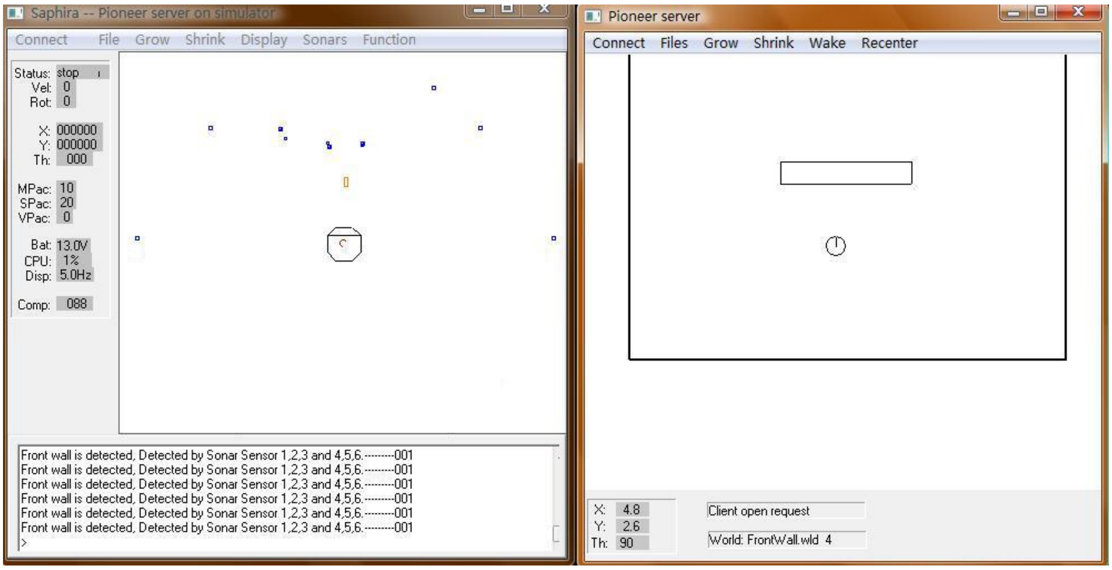


Figure 5.9 A front wall detected by six sonar sensors.

Figure 5.10 shows a situation when a front obstruction is detected by S2, S3, S4 and S5; the angle between S2 and S5 is 60°. The minimum detection width can be calculated as:

$$MD4 = 2*(ySR_3 + 250)/\tan 60^o; \qquad (5.5)$$

$$MD4 = 2*(ySR_4 + 250)/\tan 60^o; \qquad (5.6)$$

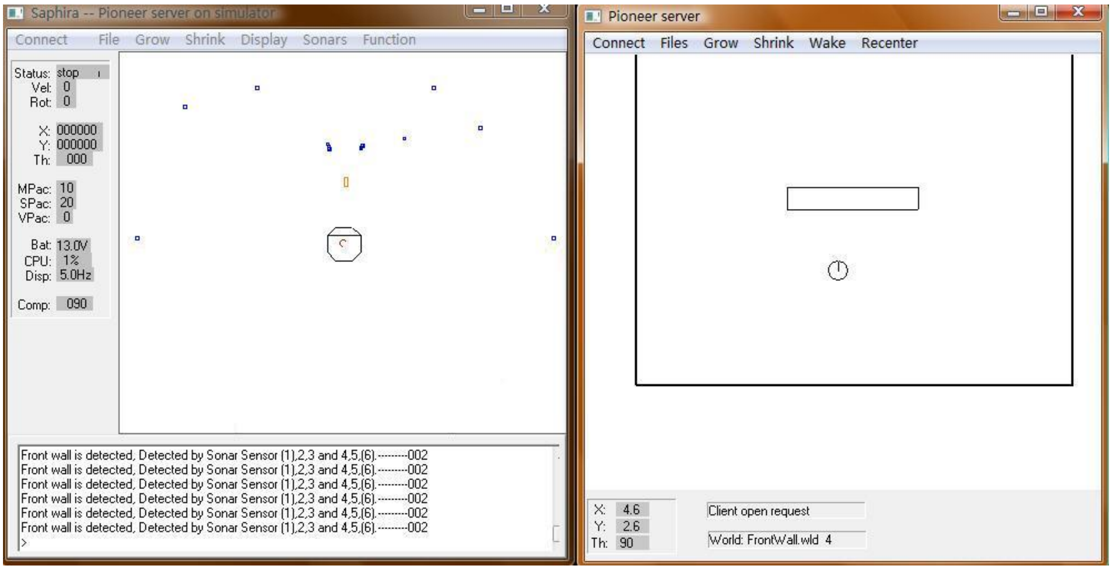


Figure 5.10 A front wall detected by sonar sensors.

The front obstruction can be classified as a front wall by three sonar sensors. Figures

5.11-5.14 show variation for front wall detection.

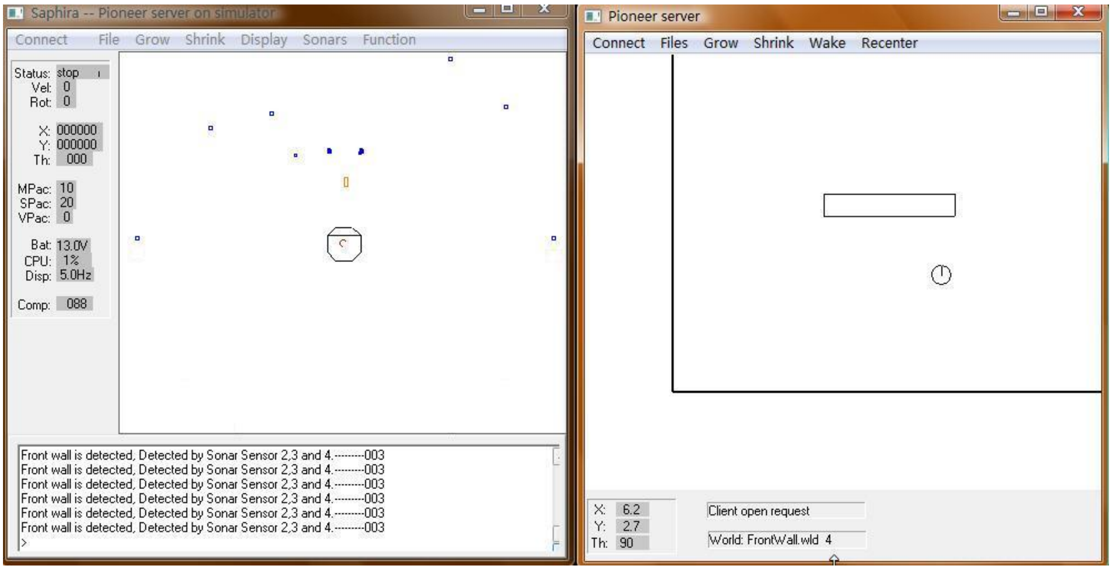


Figure 5.11 A front wall detected by S2, S3 and S4.

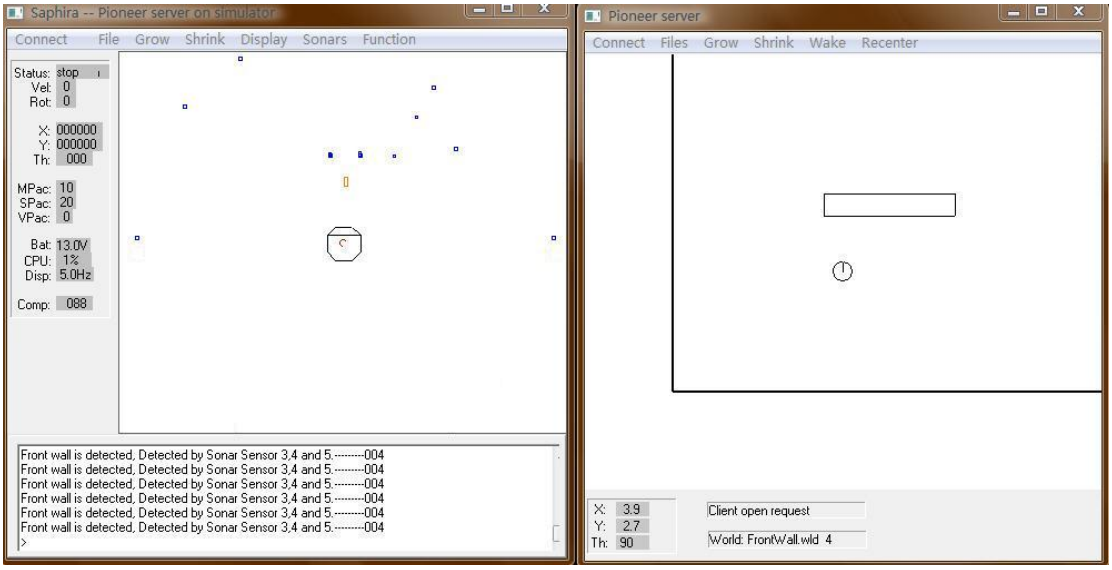


Figure 5.12 A front wall detected by S3, S4 and S5.

The angle between S2 and S4 equals the angle between S3 and S5, and the minimum detection width can be calculated as:

$$MD3= (ySR_3 +250)/\tan 60^o+(ySR_4+250)/\tan 80^o; \qquad (5.7)$$

$$MD3= (ySR_4 +250)/\tan 60^o+(ySR_3+250)/\tan 80^o; \qquad (5.8)$$

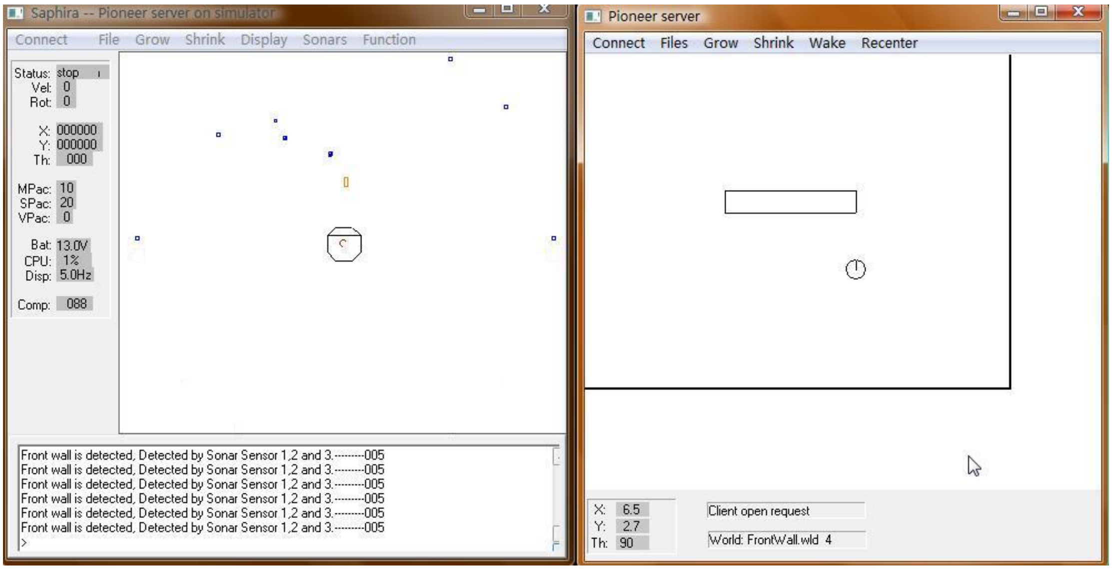


Figure 5.13 A front wall detected by S1, S2 and S3.

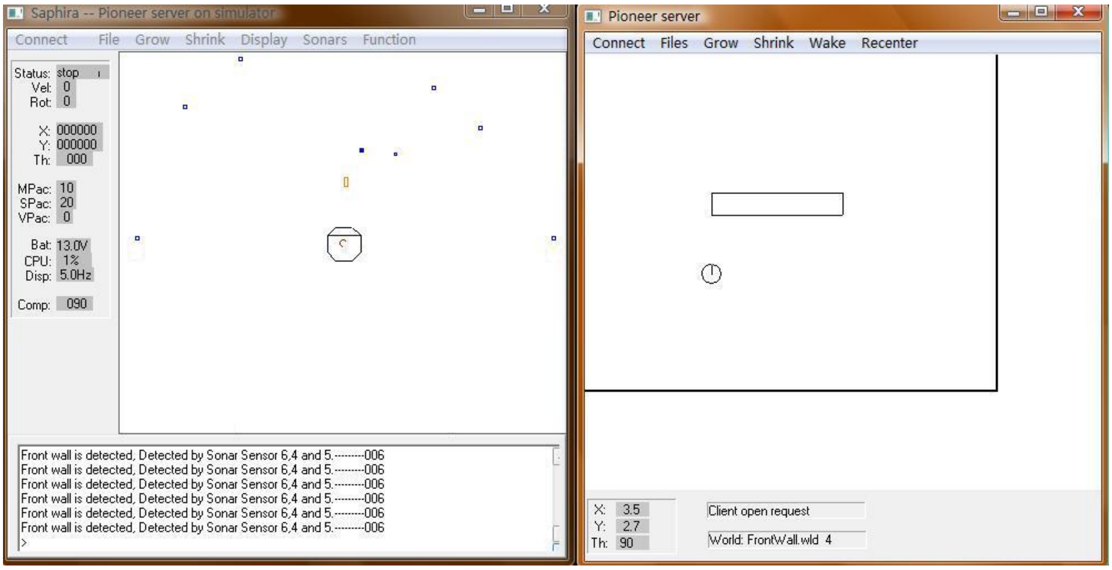


Figure 5.14 A front wall detected by S4, S5 and S6.

When the mobile robot meets a clutter of objects (see Figure 5.15), there are two sensors showing in the open range status, and three dis-neighbouring sensors showing in the detection status. As discussed in Chapter 1, Chronis' (2002, 2007) solution hence in this project rules ignore the gap which can be detected by one sonar sensor. The crucial sensor S4 is in detection status and S0 and S7 are in the open range status. These make the major votes so that it classifies the situation as a front wall. S2 and S6 provide complementary information to confirm that the obstruction is present. The left window (Figure 5.15) shows S3 and S5 are bouncing the sound wave cross the gap between the obstructions, the test program shows: *Front wall is detected by Sonar Sensor 2, 4, and 5*.

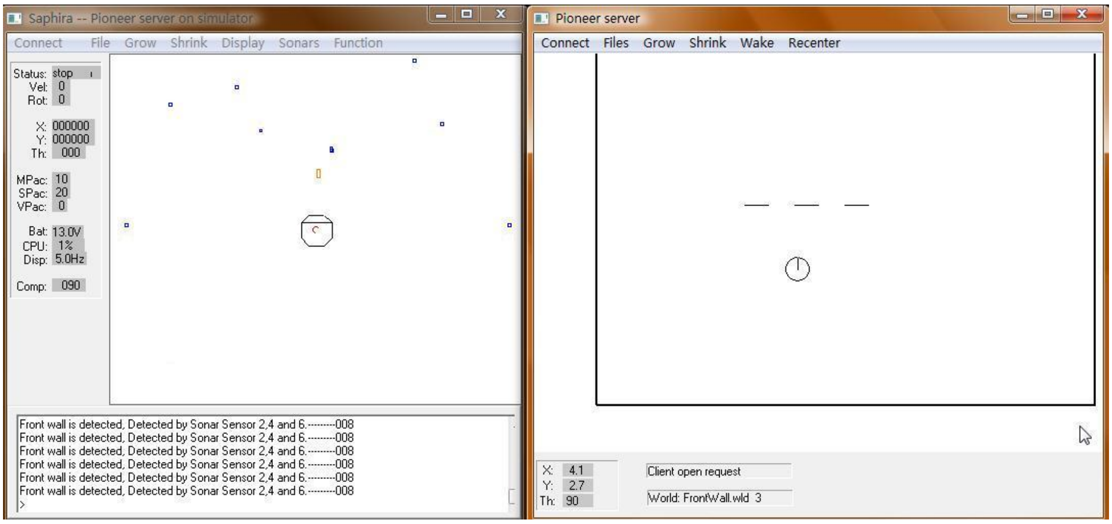


Figure 5.15 Front wall detected by S2, S4 and S6.

When sonar sensor fails, i.e., the reading of a sensor is constantly zero, the algorithm will assume the failure status as a normal detection status and use adjacent sensors information to carry on working. More information as shown in Chapter 4 will make no difference to the classification. Figure 5.16 shows the robot facing a front obstruction when S3 fails. The dots in the left window present the sonar detection points. The test program shows: *The front wall is detected by Sonar Sensors 1, 2, 3 and 4, 5, 6; Sonar Sensor 3 fails.* Figure 5.17 shows a similar situation to that in Figure 5.16 when S4 fails. Figure 5.18 shows the situation of the robot with S3 failure and detects the cluttered obstruction as a front wall.

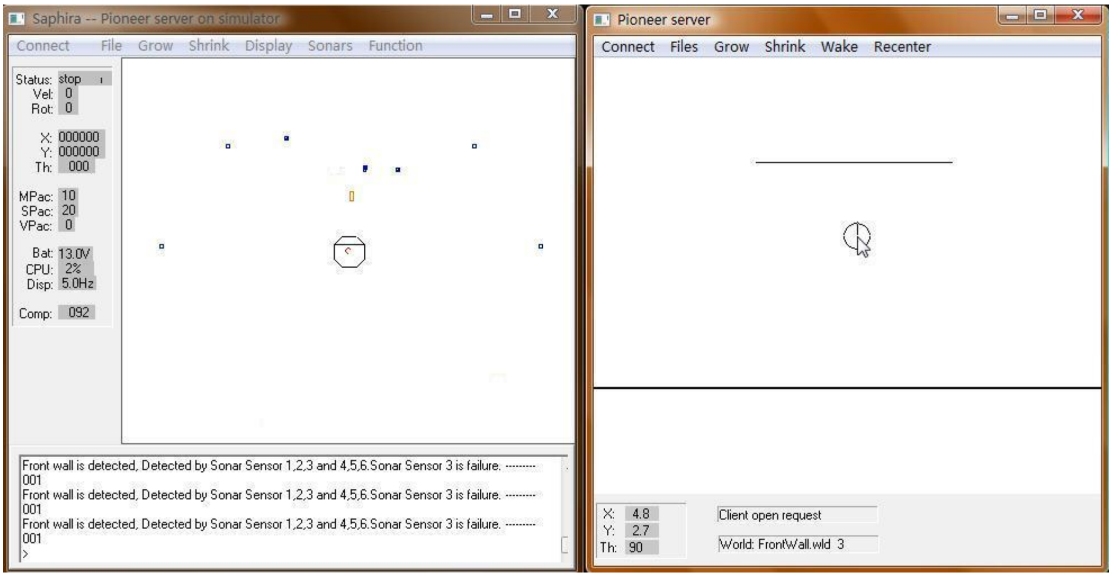


Figure 5.16 A front wall detected with S3 failure

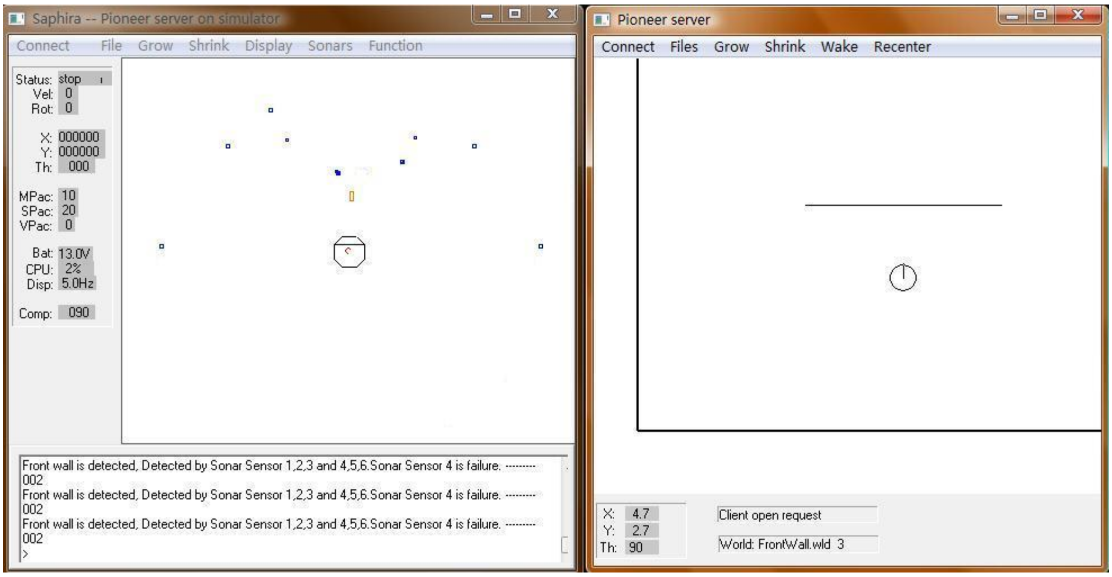


Figure 5.17 A front wall detected with S4 failure

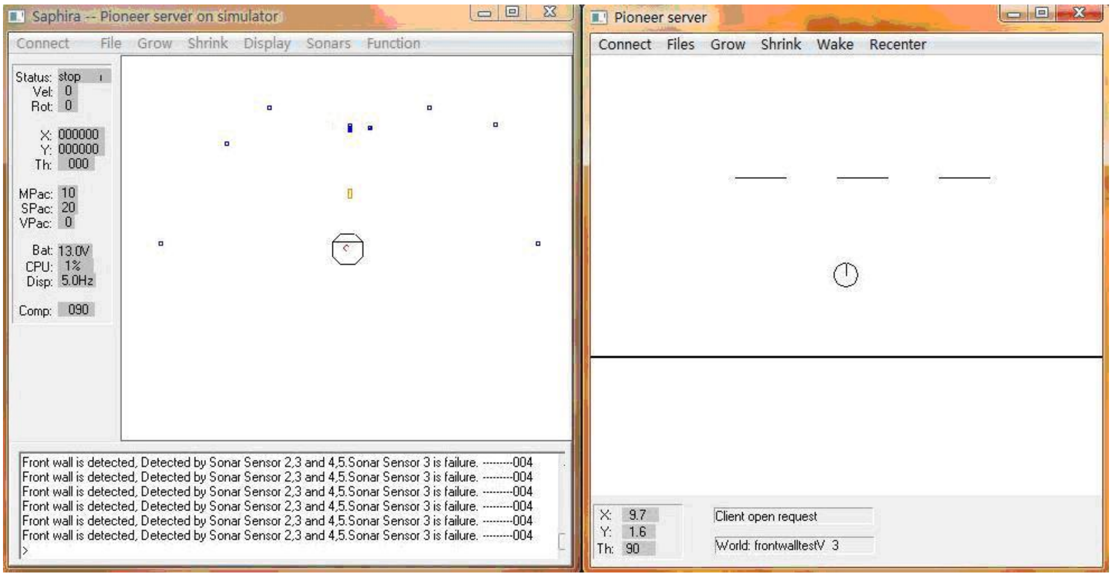


Figure 5.18 A front wall detected in a cluttered environment with S3 failure

To form a front wall /surface, in this thesis, following conditions must be satisfied:

- The reading of S0 and S7 must be in the open range status, which falls into the returning the maximum range.
- At least either S3 or S4 is in the detection status.
- No adjacent sonar sensors fail or in the open range at the same time.

The main problems in sonar responses are:

- Angular uncertainty, where the uncertainty in the angle information of a sonar response from a detected object.
- Specular reflection, which refers to the sonar response that is not reflected back directly from the target object. In a specular reflection, the ultrasound is reflected away from the reflecting surface, which results in longer range reporting or missing the detection of an object.

These problems can be found from the figures, Figure 5.17 for example, the dots in the left window are the sonar detection points, and S1 and S2 are clearly getting specular returns. During the detection, the scheme takes the detection status of sonar sensor into account, so the scheme has innate attributes to tolerate specular reflection and angular uncertainties.

### 5.4.3.2 Detection of Side Walls

When detecting a left wall, it is required that the reading of S0 must be smaller than the maximum reading. It has already been stated that it takes three neighbouring sensors to form a surface. Figure 5.19 shows the situation when only S0, S1 and S2 are shorter than the maximum reading and the scheme detects the situation and takes it as a left wall. Figure 5.20 shows that the robot detects a right wall.

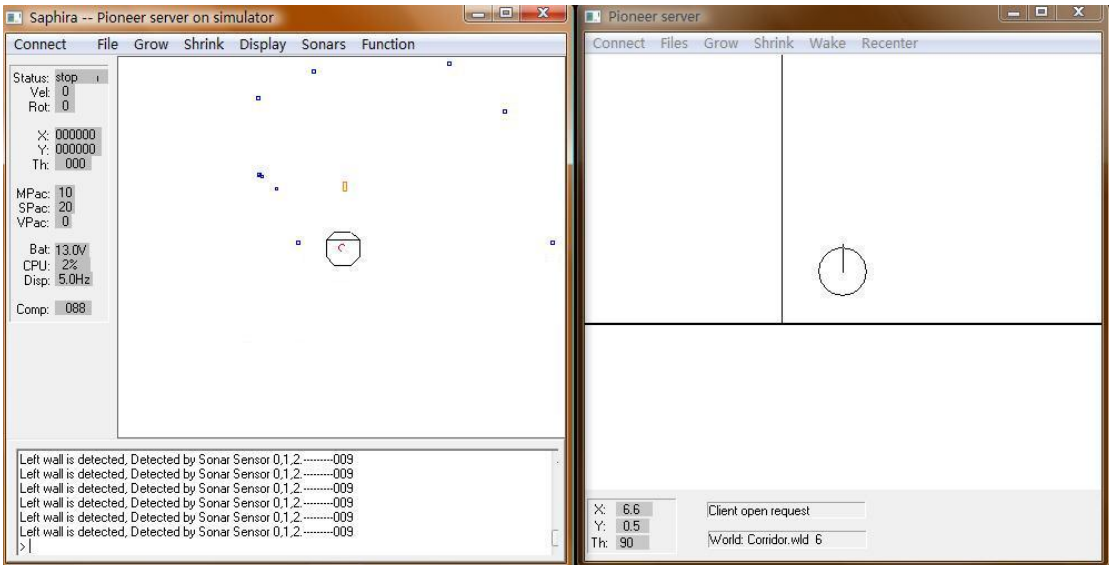


Figure 5.19 A left wall detected by S0, S1 and S2.

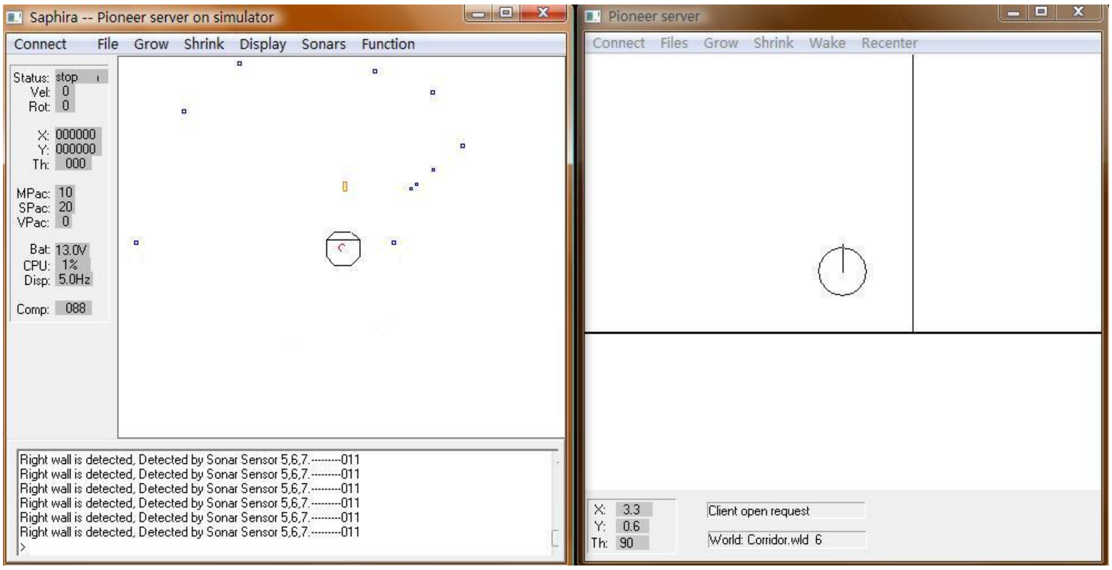


Figure 5.20 A right wall detected by S7, S6 and S5

Therefore, to form a left wall the following conditions must be satisfied:

- The readings of S0, S1 and S2 return shorter than the maximum reading.
- The readings of S7, S6, S5 and S4 return open range.

Based on the same principle to form a right wall in a structured world the following conditions must be satisfied:

- The readings of S7, S6 and S5 return shorter than the maximum reading.
- The readings of S0, S1, S2 and S3 return open range.

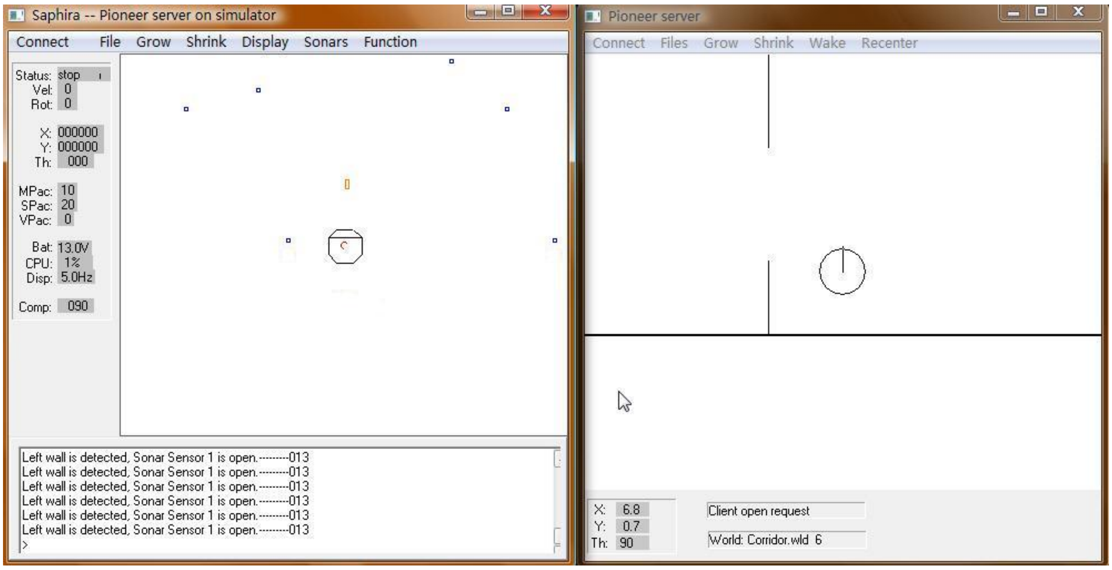


Figure 5.21 A side wall detected by S0 and S2

When the mobile robot meets a structure shown in Figure 5.21, the critical sensor S0 is engaged and S2 is providing the complementary information to confirm that it is a dis-neighbouring obstruction. The situation shown in Figure 5.21 is classified as a left wall.

To form a left wall in a cluttered world, the following conditions must be satisfied:

- The readings of S0 and S2 must return shorter than the maximum reading.
- The readings of S7, S6, S5 and S4 return open range.

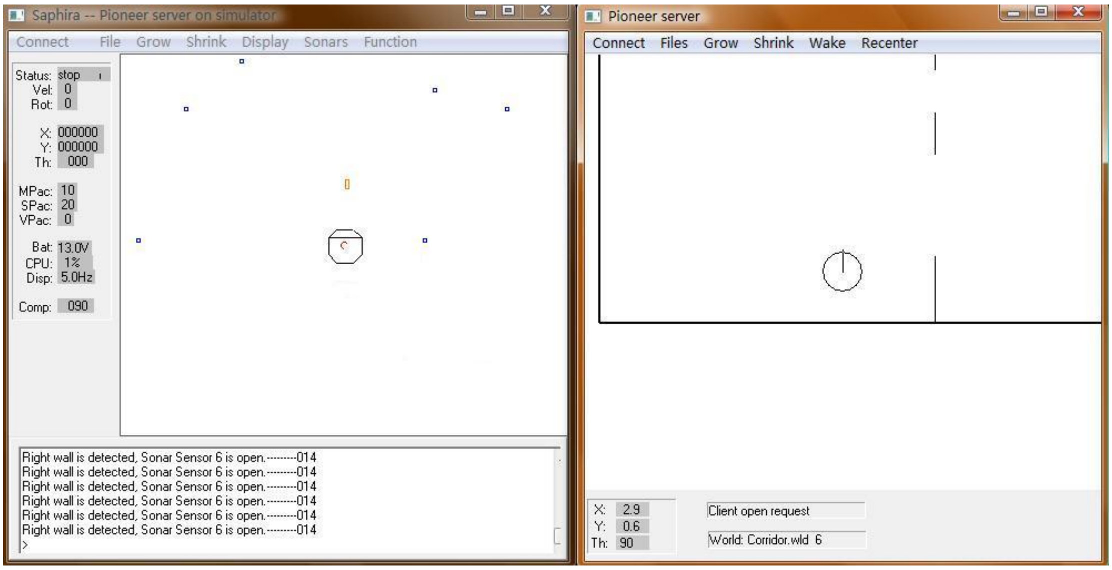


Figure 5. 22 The side wall detects by S5 and S7.

To form a right wall in a cluttered world, the situation must satisfy the following conditions, (see Figure 5.22):

- The readings of S7 and S5 must return shorter than the maximum reading
- The readings of S0, S1, S2 and S3 return open range.

Figure 5.23 shows the situation when the robot detects a left wall with S1 failure. Figure 5.24 shows the situation when the robot detects a right wall with S6 failure. Figure 5.25 shows the situation when the robot in a cluttered environment detects a left wall with one sensor failed.

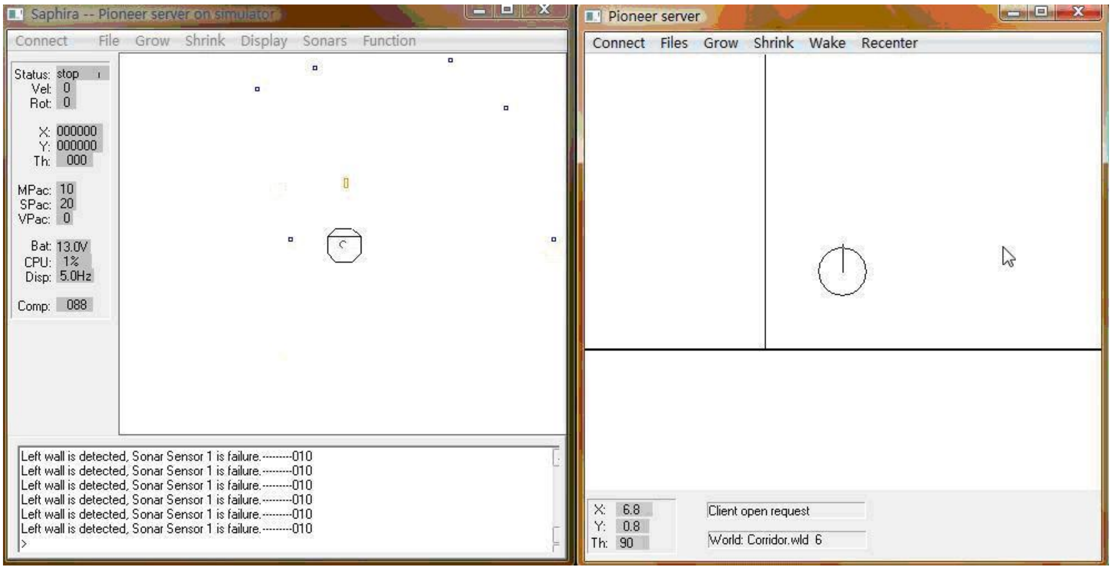


Figure 5.23 A left wall is detected when S1 is failure.

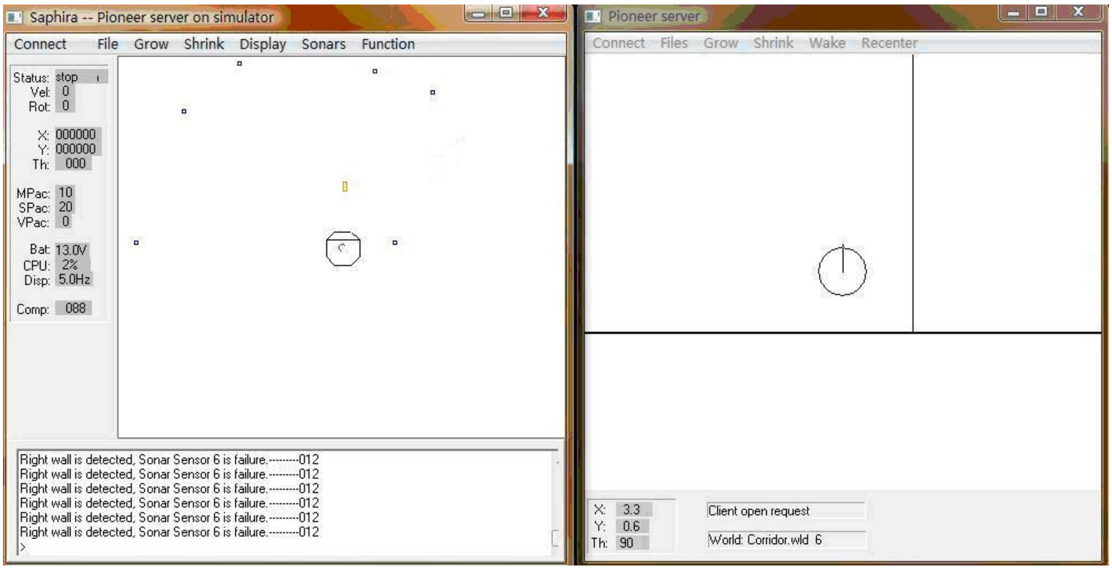


Figure 5.24 A right wall is detected when S6 is failure.

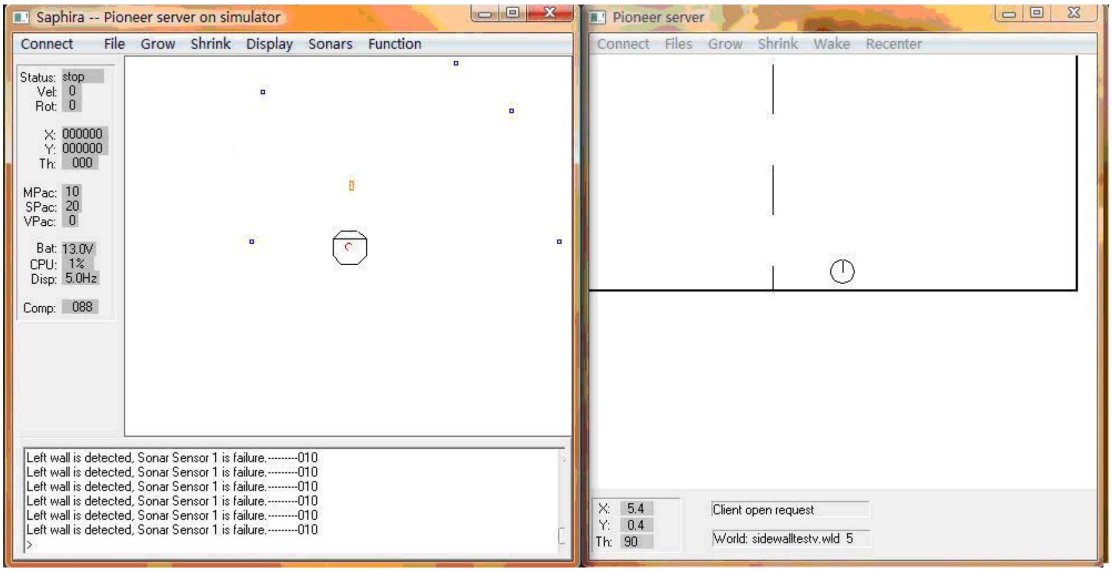


Figure 5.25 A left wall with S1 failed in a cluttered environment.

### 5.4.4 Detection of Corridor

Figure 5.26 shows the mobile robot travelling along a corridor, where there are doors or gaps, but the left part of the figure shows all the sensors except for S3 and S4 are in the detection status. The corridor can be assumed as a left wall and a right wall. Thus, when the robot detects a corridor, it can be thought of as detecting of a left wall and a right wall at the same time. Note the corridor in Figure 5.26 is made up by one left wall and one right wall in a cluttered environment; see Figures 5.21 and 5.22. In Figure 5.26, the left and the right wall are dis-neighbouring objects. In the left window S0, S1, S2 and S5, S6, S7 are all engaged, and the scheme classifies the situation as a corridor.

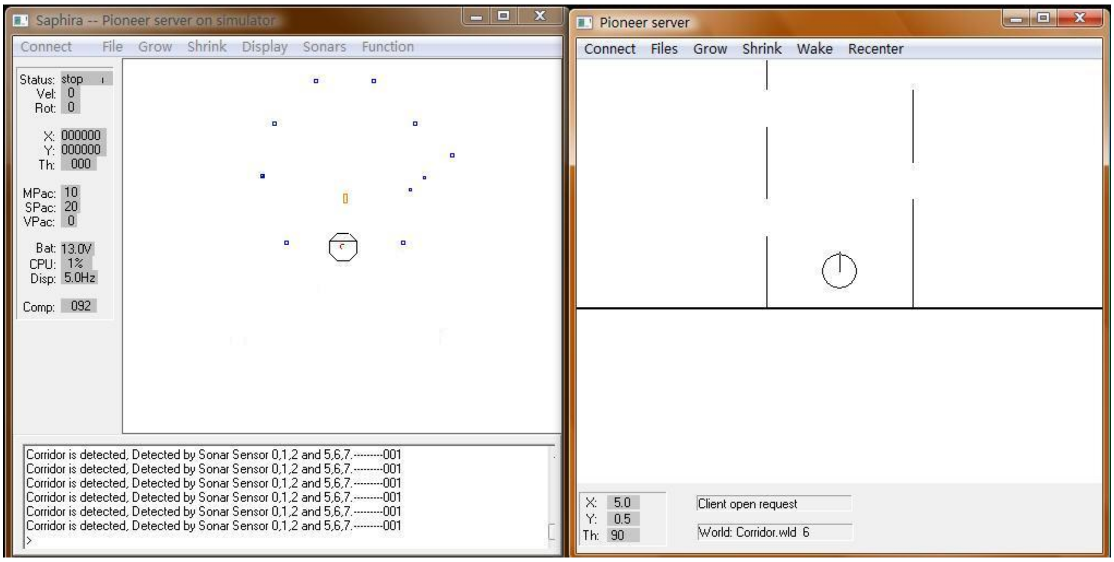


Figure 5.26 A corridor detected

To form a corridor, the following conditions must be satisfied:

- The readings of S0, S1, S2, S5, S6 and S7 are shorter than the maximum reading.
- S3 and S4 are in the open range status.

As in the previous discussion about the tolerance to the failure of sensors, the scheme assumes the failed sensors are in normal detection status; in the situation of sensor failure the same conditions must be satisfied, Figures 5.27- 5.30 show various situations of tolerance to failures.

The situation of detecting a corridor in a cluttered world can be viewed as a cluttered left/right wall detected by the mobile robot, figures 5.31- 5.34 show a variety of situations. To form a virtual corridor the following conditions must be satisfied:

- The readings of S0, S2, S7, S5 return shorter than the maximum reading.
- S3 and S4 are in the open range status.

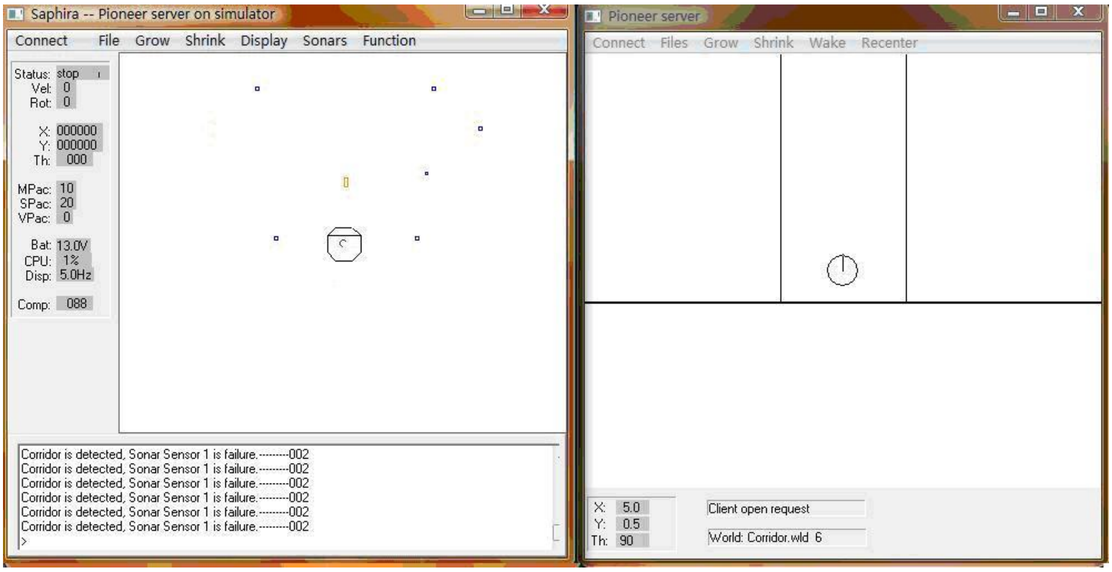


Figure 5.27 A corridor is detected with S1 failure.

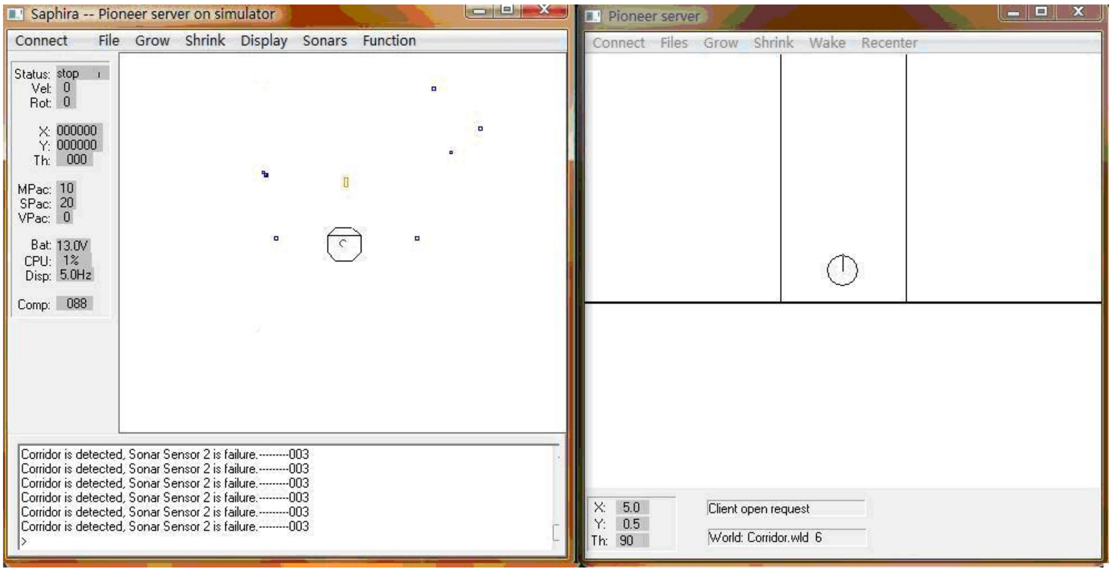


Figure 5.28 A corridor is detected with S2 failure.

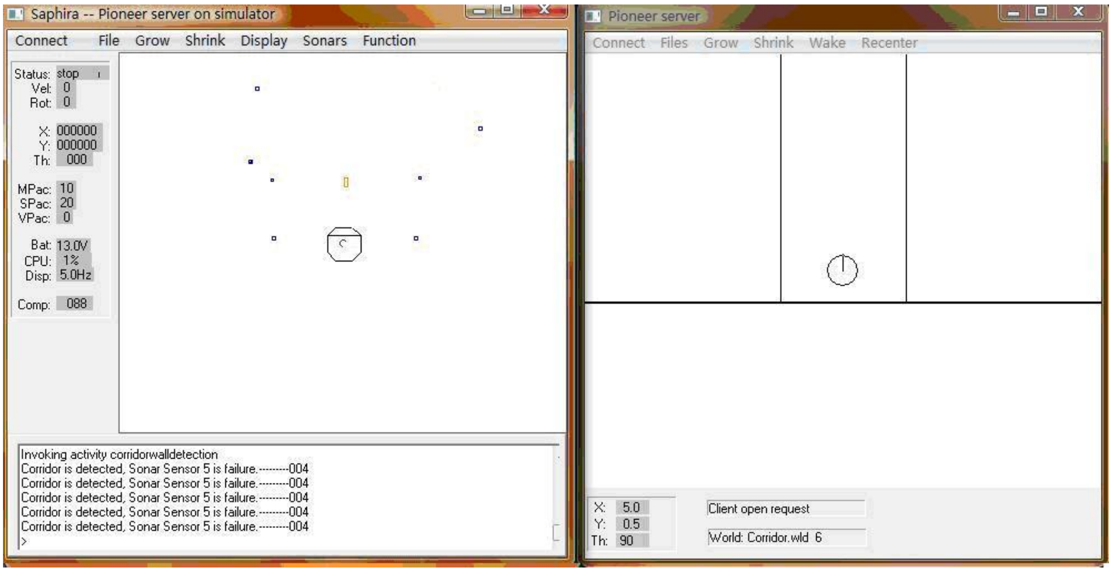


Figure 5.29 A corridor is detected with S5 failure.

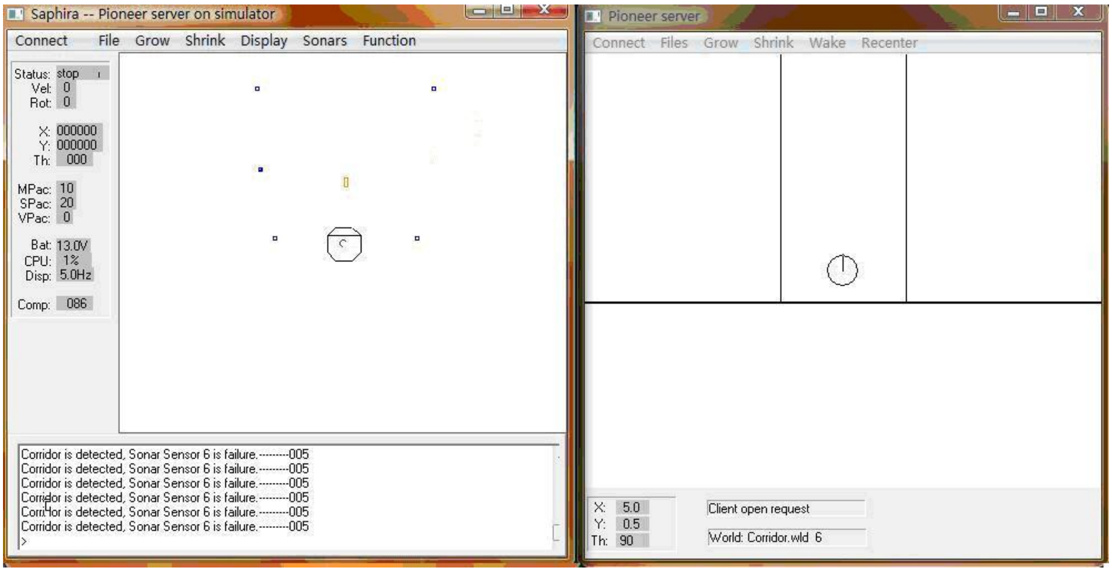


Figure 5.30 A corridor is detected with S6 failure.

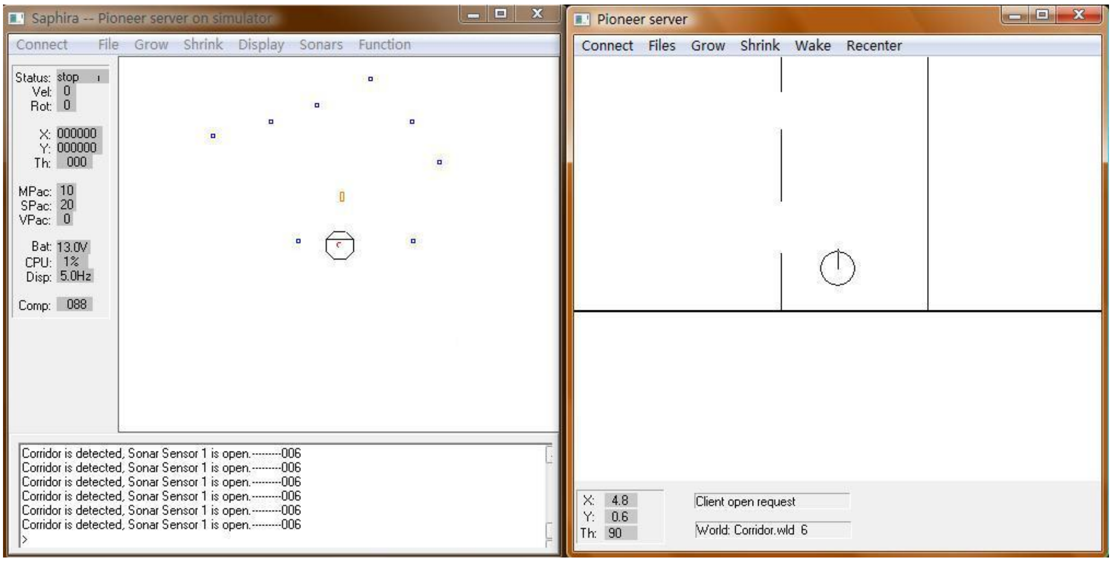


Figure 5.31 A cluttered type corridor is detected with S1 in the open range.

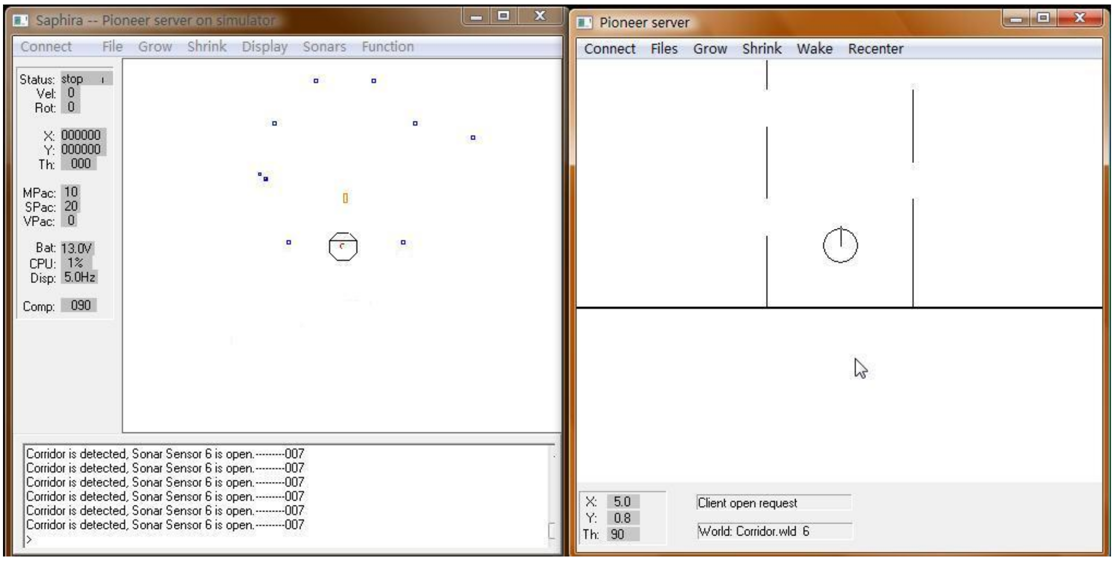


Figure 5.32 A cluttered type corridor is detected with S6 in open range.

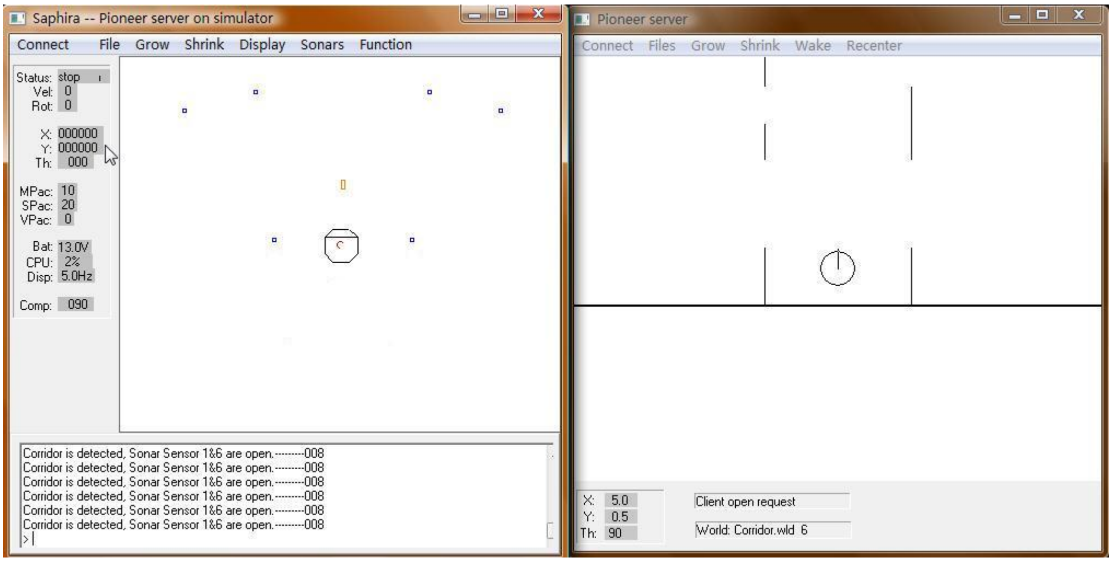


Figure 5.33 A cluttered type corridor is detected with S1 and S6 in the open range.

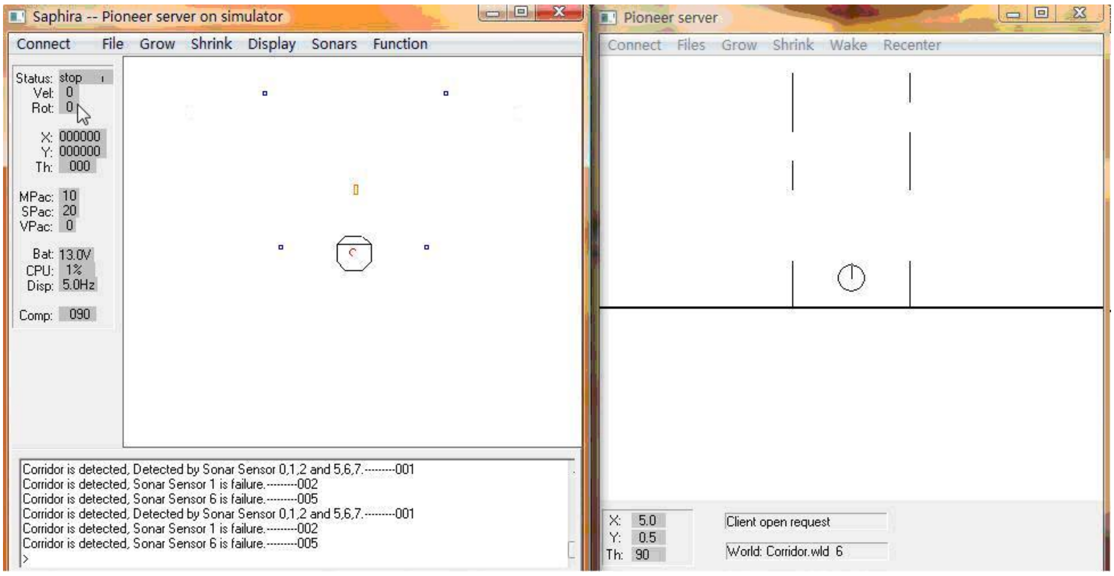


Figure 5.34 A cluttered type corridor is detected with S1 and S6 failures.

### 5.4.5 Detection of a Corner

A left hand corner can be taken as being composed of a left wall and a front wall. To form a left wall, it requires three sensors to detect the obstruction. However, the readings of S1 and S2 may not be gained by the reflection from the left wall but from the front obstruction when the robot closes to the front obstruction, see Figure 5.35. The scheme treats the situation as a left wall as long as the required detection conditions are met.

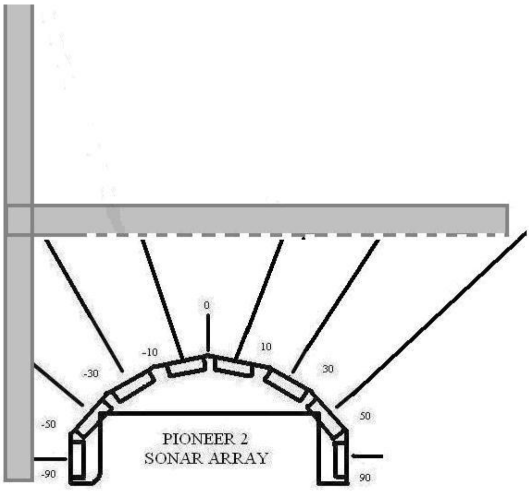


Figure 5.35 Distinguishing a side wall and a corner

To form a left hand corridor, the situation must satisfy the conditions below (Figure 5.36):

- The reading of S7 must return the maximum reading.
- And the readings of any neighbouring sensor combinations which including S0 and S1 return shorter than the maximum reading.

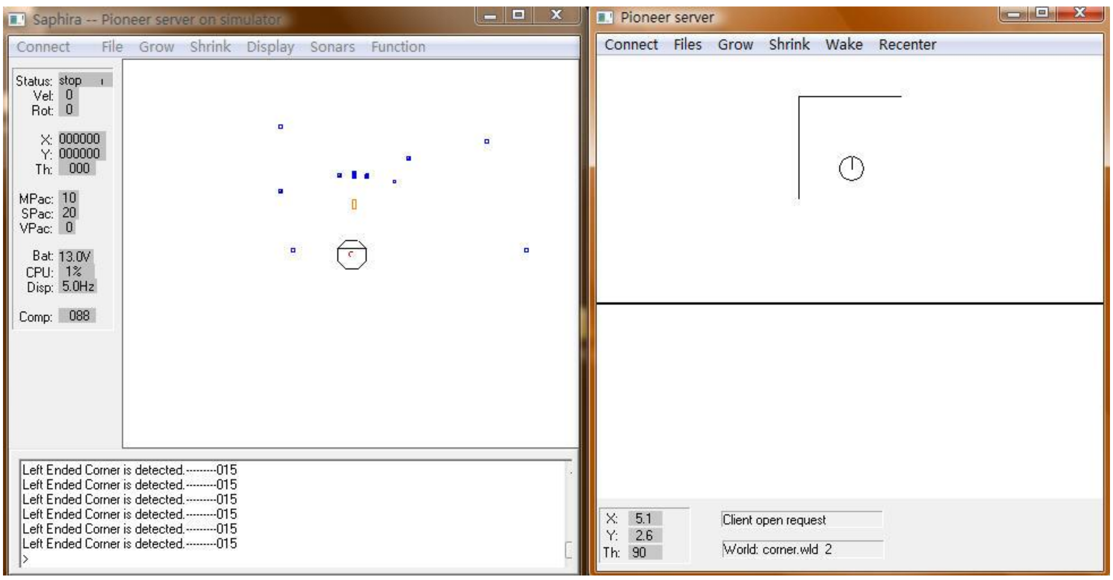


Figure 5.36 Detection of a left-hand corner.

Using the same principle for forming a left hand corner to form a right hand corner, the following conditions must be satisfied (see Figure 5.37):

- The reading of S0 must return the maximum reading.
- And the readings of any neighbouring sensor combinations which including S7 and S6 return shorter than the maximum reading.

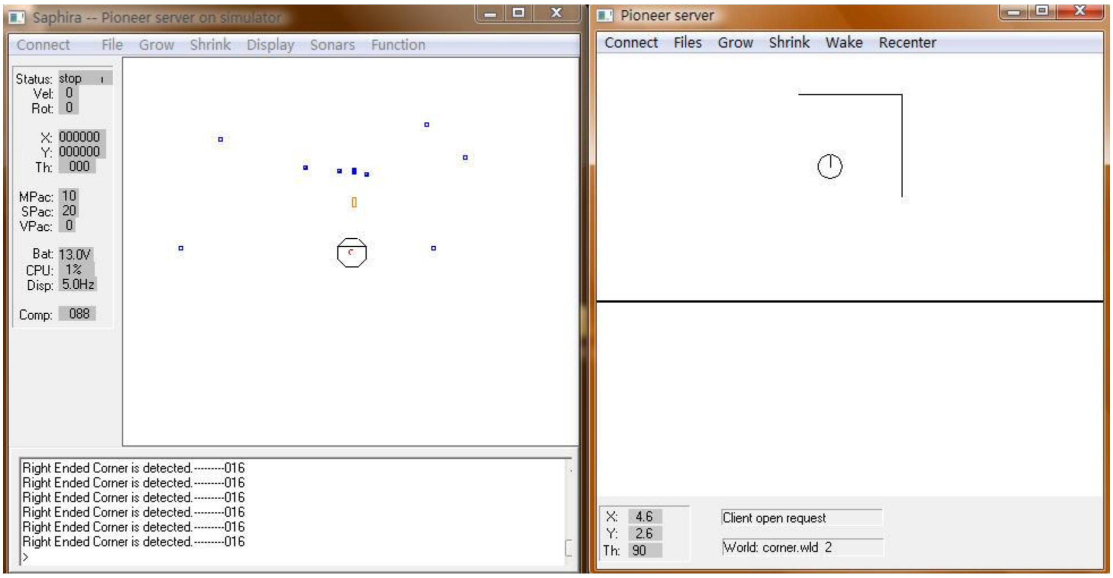


Figure 5.37 Detection of a right-hand corner.

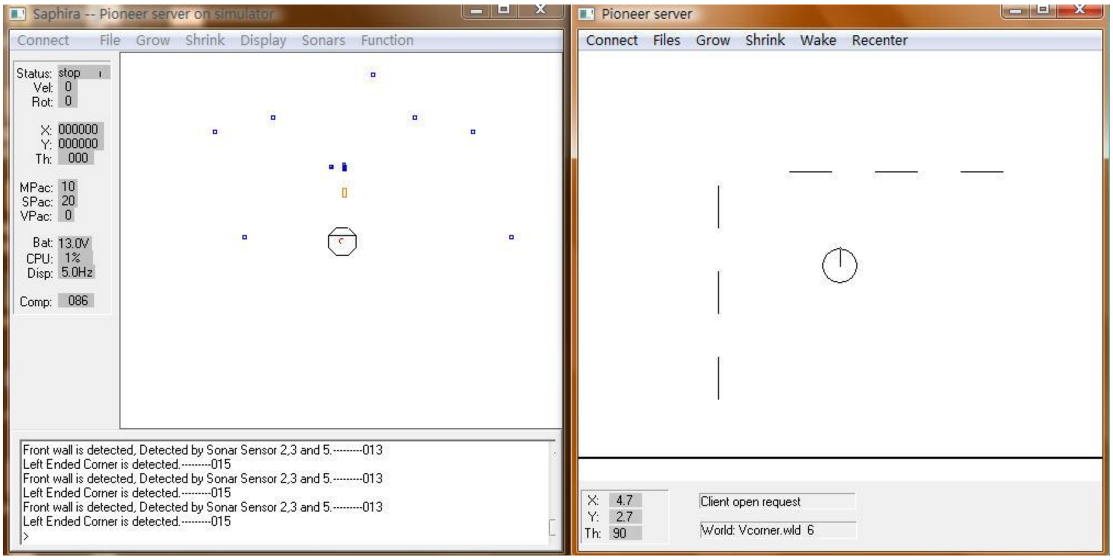


Figure 5.38 Detection of a left-hand cluttered corner by S2, S3 and S5.

In the situations of Figure 5.36 and 5.37, Chronis' approach thinks these tow type of corners as two separate objects located vertically; also it indicates the robot can go through the gap between the objects. This decision will cause the robot collision.

When the mobile robot travels among the cluttered objects, the situation shown in Figure 5.38 can be classified as a cluttered left hand corner. In other perspective view, the Chronis' approach detects one object on the left which detects by S0; it also detects two objects ahead the robot, and indicates the robot can go through the gap between these objects. The decision is misled by S4, because sonar bouncing through the gap.

To form a left hand corner in a cluttered situation, the following conditions must be satisfied:

- The reading of S7 must return the maximum reading.
- The reading of S0 must return shorter than the maximum reading.
- The reading of S3 or S4 must return ones that are shorter than the maximum reading.
- The reading of two neighbouring sonar sensors cannot return the maximum reading at the same time.

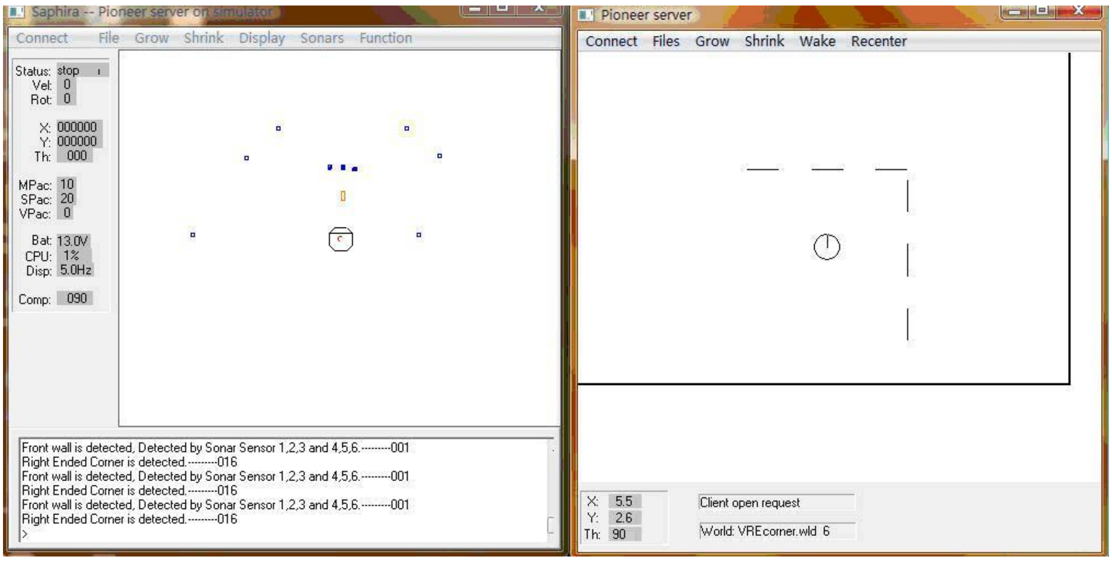


Figure 5.39 Detection of a right-hand cluttered corner.

To form a right hand corner in a cluttered situation the following conditions must be satisfied:

- The reading of S0 must return the maximum reading.
- The reading of S7 must return one is shorter than the maximum reading.
- The reading of S3 or S4 must return one that is shorter than the maximum reading.
- The reading of two neighbouring sonar sensors cannot return the maximum reading at the same time.

### 5.4.6 Detection of a Dead-end

When the mobile robot meets a dead-end, mobile robot is blocked in any detection direction, shown in Figure 5.40. To form a dead-end, it requires following conditions:

- The readings of S0 and S7 must return shorter than the maximum reading.
- The reading of S3 or S4 must return one that is shorter than 1300mm.
- Not any two neighbouring reading of sonar sensors both return the maximum readings.

The dead-end can be considered as a block made up jointly by a corridor and a front wall. To ensure making a good decision, there is an extra condition to form a dead-end, that is, the distance to the front wall must be shorter than 1300mm.

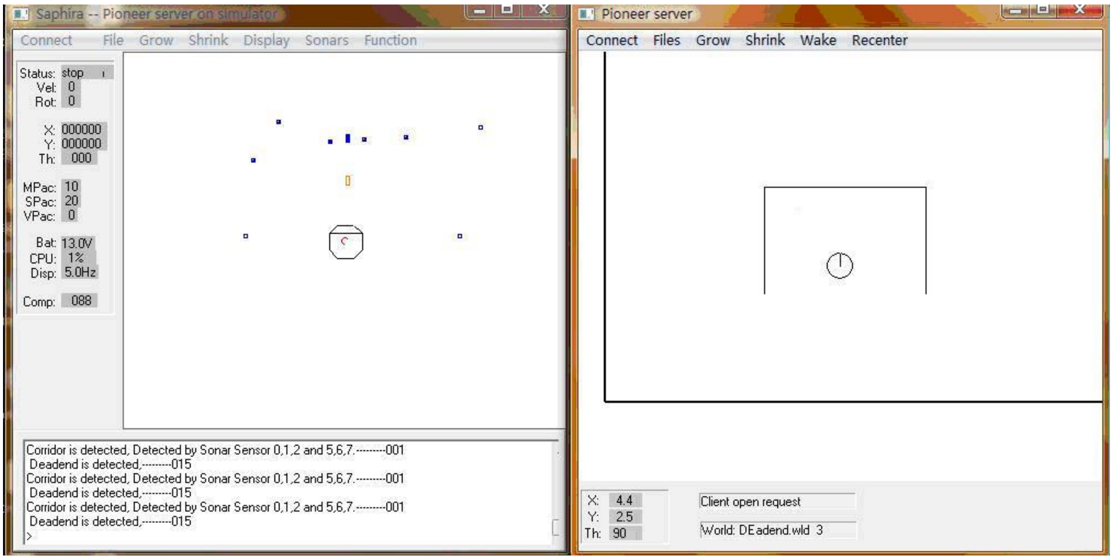


Figure 5.40 Detection of a dead-end.

### 5.5 Conclusion

To test and verify the rules developed, it requires the running of the robot in various situations. There are two types of obstacles. One is detected by a single sonar sensor, and the other is detected by two or more adjacent sensors. Later in Chapter 7 what will be seen is that when the robot meets multiple obstacles located around it, the strategy needs a small modification.

In this chapter, the sufficient condition is also developed for the detection of different types of surfaces like a front wall, left wall and right wall. It is found that such surfaces require at least three adjacent sensors to be engaged at the same time. In a cluttered situation, when the middle sensor out of the three adjacent sensors is not engaged, a “virtual surface” is created by utilizing the reading(s) from the other two sensors. It is observed that the algorithm overcomes the uncertainty associated with the specular return of sonar sensors. The strategy also has an element of tolerance to faults, a feature that requires defining certain rules of failing sensors.

In later chapters more complex surfaces and objects are detected, and what is also shown is how corners can be decomposed into their primitives in terms of surfaces e.g.

- The corner can be taken as being made up vertically together by a front wall and a side wall. There are two types of corners: left hand and right hand corners. In addition, the acute angled and obtuse angled corners are also tested in Chapter 7.
- Two side walls that parallel to each other can be considered as a corridor.
- A dead-end can be viewed as a corridor together with a front obstruction.

During the progress of testing and verification, the definitions and limitations of each type of structures are determined. After the recognition of the structures, the strategies of avoiding these structures are introduced and tested in Chapter 6.

# Chapter 6 Online Avoidance Strategies: An Intuitive Quadrant Approach

## 6.1 Introduction

In Chapter 5, it was shown that the classification rules are able to generate structural information regarding its local environment by using sensor readings and geometric information, which is then utilized to generate a local plan to avoid obstacles and to move towards the goal. However, such movements, which are dependent on the sensor readings, do require bounds to be placed so that the robot does not get too close to the various obstacles and surfaces. Indications of these were given in Chapter 5, where the maximum and minimum widths of an object were given.

The online obstacle avoidance controller evaluates the situation of a mobile robot, and controls all the movements and makes decisions in order to achieve its goals. During avoidance stage, the online controller will control the movement of the mobile robot in order to avoid obstacles of any form, and hence also to avoid collision. The robot uses information from the sensors to build a local structure of the environment around it, see Chapter 3. It also selects the mobile robot's actions between the direct control and behaviour control. It has rules and strategies for avoiding each type of structures, (see sections 6.5-6.8). When the rules classify the environment into structures, the controller desires strategies to drive the mobile robot so that it avoids the obstruction. To do this, quadrant system is used so that the controller can determine which quadrant the goal point lies in. In the process of structure-avoiding, the rules compare with the current robot location to determine movement of the robot.

In this chapter a detailed obstacle avoidance strategy is developed and implemented so that the robot can perform its task online and in real-time. Furthermore some special situations are also introduced.

## 6.2 The Safety Parameter

When the robot moves close to the obstacle, one question is raised: when does the robot stop – when the sonar reading is 0 mm, or 50 mm, or…? At the same time the scheme need to consider the size of the robot, the location of sensors on the robot, and the ability of the sensor to detect an obstruction of whatever size within the limitations discussed in Chapter 5. This implies the need for a safety parameter, which indicates the safety distance between the robot and obstruction. Thus the robot that comes to a stop much before it hits the obstacle. In Chapter 3, it is mentioned that the robot has 3- classes' sensors, namely, left, right and front detection sensors. Thus rather than having a safety parameter for each sensor, a safety parameter is devised for each of these classes of sensors. The technique sets up each safety parameter in front, left and right directions, see Figure 3.6. For example we set the safety parameters for left and right sides as 200mm, which can use the reading of S0 and S7 to measure the distance. As sensor S0 is the main detection sensor in the left direction, when the $SR0 < 200mm$ the mobile robot must halt immediately.

## 6.3 Quadrant System

A quadrant system approach is designed in the egocentric view of the robot. In the quadrant system, the robot itself is taken as the origin, and the quadrants are all relatively to the robot, see Figure 6.1.

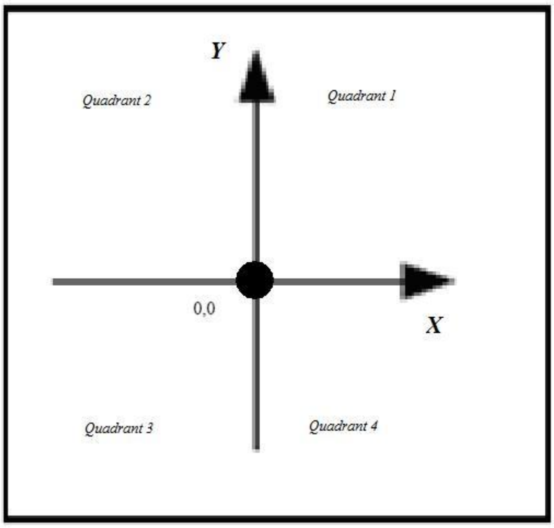


Figure 6.1 The quadrant system with the robot as origin (the black dot).

The initial robot coordinates is (0, 0), and the quadrant system requires online updating of the robot' current global position by calling the function in Saphira. When the mobile robot detects a structure, the scheme determines whether the structure will cause obstruction to the mobile robot or not. For example, when the mobile robot moves towards a front wall and the goal point is in Quadrant one, the mobile robot will move along the front wall from left to right. If the goal point is in quadrant, two the mobile robot will avoid from the right to left. The front wall may not make an obstruction to the mobile robot, when the goal point is in Quadrant three or four. The current global position of the mobile robot can be determined by embedded odometry system in Saphira. The current coordinates of the robot can be presented as (RobotX, RobotY), and the coordinates of the goal point are determined by the user (GoalX, GoalY). To calculate the quadrant of goal point:

*If (GoalX-RobotX)>0, and (GoalY-RobotY)<0, then the goal locates in quadrant one;*

*If (GoalX-RobotX)>0, and (GoalY-RobotY)>0, then the goal locates in quadrant two;*

*If (GoalX-RobotX)<0, and (GoalY-RobotY)>0,then the goal locates in quadrant three;*

*If (GoalX-RobotX)<0, and (GoalY-RobotY)<0, then the goal locates in quadrant four.*

Table 6.1 The method for determining goal quadrant.

## 6.4 The Obstacle Avoidance Strategy.

In Saphira, there are two ways to control robot motion. The direct motion control is appropriate for moving the robot with simple sequences of action. However in certain cases, the trajectory of the robot must satisfy complicated demands from the task and various maintenance policies; the complex control is to decompose the problem into a set of small actions to accomplish particular goals, which can then be combined into a more comprehensive control strategy. Each such small action, with its associated goal, is called a *behavior*. A behavior looks at some set of sensor information and outputs a desired action, based on its goal. In the situation of navigation, the mobile robot moves from one point to a desired goal point. The rules use *sfAttendAtPos* behavior to achieve the nominal path which is a straight line from the starting point to the goal point. The behavior *sfConstantVelocity* maintains the speed of the mobile robot during it moving and *sfStop* stops the robot when the goal is achieved. The example is the file for determining the goal and the set up of the behaviors.

```
sfInitControlProcs();        /* for behavior control */
sfInitRegistrationProcs(); /* register the robot using sensed artifacts */
sfInitInterpretationProcs();     /* find walls and doors */


point *goal;
goal = sfCreateGlobalPoint(8000,-6000,0);
sfAddPoint(goal);
start sfConstantVelocity(200) priority 2 suspend;
                start sfAttendAtPos(200,goal,100) priority 3 suspend;
start sfStop priority 4 suspend;
```

Table 6.2 Example of setting up the goal point in Colbert

Behaviours are processes or control laws that can achieve and/or maintain goals. The goal of the system is achieved by subdividing an overall task into small independent behaviours or activities that focus on execution of specific subtasks [Seraji & Howard (2002), Jaafar & McKenzie (2008)]. Unfortunately it is difficult to make good decisions to satisfy both goals and constraints. [Saffotti (1998)]. As a result, behaviour or activity rules conflict with one another, meaning that more than one rule becomes active at one time. No perfect action selection mechanism has been developed, since different systems have different requirements [Humphrys (1997)].

The action selection management in this thesis involves local avoidance activities for different types of structure and several behaviours. For example, the ***sfAttendAtPosition*** achieves the goal of path planning. The behaviour controls the heading and movement from starting point to goal point. It is the nominal path, a straight path from start pointing to goal point. During the global path planning and local avoidance stage, the behaviour ***sfConstantVelocity*** maintains the mobile robot travelling in a steady and safe speed. The ***sfStop*** stops the mobile robot when the goal is achieved. During the travel along the nominal path generated by ***sfAttendAtPositio***n and when the obstruction appears, the scheme classifies the obstruction as structures and suspends ***sfAttendAtPosition***; local avoidance strategies will take over the control, and details of each individual local avoidance strategy will be introduced in later section. After the robot is away from the obstruction and in a clear area, the algorithm resumes ***sfAttendAtPosition***. The algorithm suspends all detection activities when the mobile robot one meter away from the goal point and remains ***sfAttendAtPosition, sfConstantVelocity, sfStop*** for the final stage to achieve the goal.

When the mobile robot classifies the obstruction as an obstacle in travelling, sensor S3 and S4 are the heading of the mobile robot. The scheme will suspend the global path planning and perform the local avoidance task when the route is blocked. S3 and S4 are both critical detection sensors for detecting front obstructions. For example, the obstruction is detected by S3 or S4, the obstruction may block the robot's heading direction and it may also block the mobile robot's travelling route. When only S3 is detecting an obstruction, the mobile robot will decide to avoid the obstacle by steering right, and keep travelling when the mobile robot moves to an open area. It will be towards to the goal again. The same concept is used for the obstruction detected by S4. The mobile robot will make a decision to avoid the obstacle by steering left. When the mobile robot detects the obstacle on the left or right side which does not interfere with the path, the scheme will allow the robot to follow the nominal path to the goal. For example, in Figure 6.2, the goal point coordinate is (6000, -500), and the robot travels towards to the goal point. When the obstacle is detected by both S3 and S4, it acts until the obstacle is close enough, i.e. the sonar sensor detection range is less than 1300mm. This confirms that the classification of obstruction as a single obstacle. And according to the equation 6.1, it can estimate the width of the obstruction by substituting the SR3 value, see also Chapter 5.

$$MD2 = 2*(ySR_3 +250)/tan70^o; \qquad (6.1)$$

The scheme steers the robot to the right, and assigns a new nominal path. The robot travels on the nominal path and detects the object by S1 and S2. The object is no longer on the nominal path. The robot stays on the path to achieve the goal.

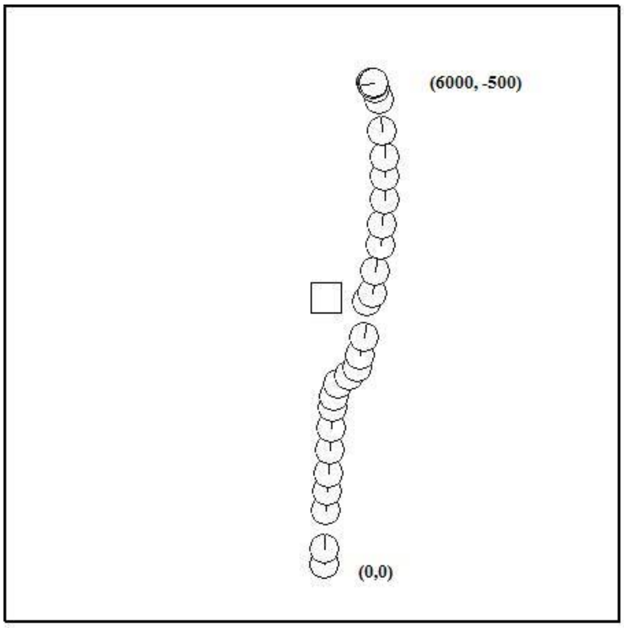


Figure 6.2 the mobile robot avoiding an obstacle.

### 6.5 The Strategy for Avoiding Surfaces.

From the robot's egocentric point of view, there are three surface structures: a front wall, a right wall and a left wall. In Figure 6.3, the goal point is set up at (8000, -1000) which is the point the trajectory ends. The robot meets the obstruction, once the obstruction is confirmed as a front wall; the scheme suspends the global path planning task. The goal point is in the second quadrant and the scheme ensures that the robot avoid the obstacle from the right. After turning right, the robot moves parallel with the front wall, and this situation becomes one where the robot detects a left wall. In the situation of avoiding a left wall, it takes to know where the goal point is and whether the left wall makes an obstruction between the goal point and the robot.

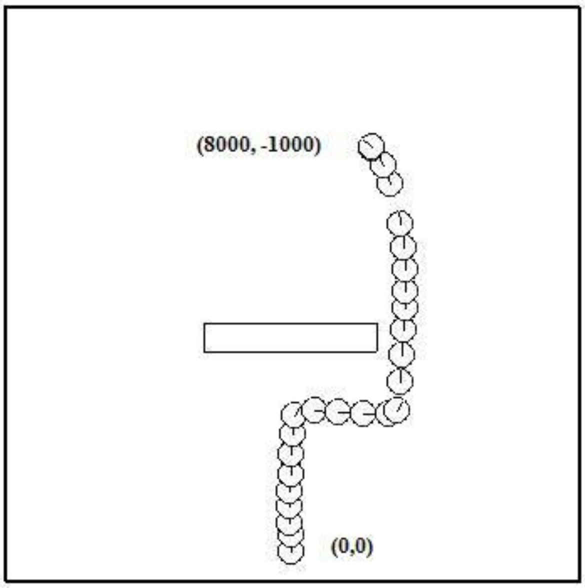


Figure 6.3 The mobile robot avoiding a front wall.

The quadrant system monitors which quadrant the goal point is in. It requires to keep tracking the heading of the robot for ensure that the side wall is avoided. Figure 6.4 shows the interpretation of the mobile robot heading system.

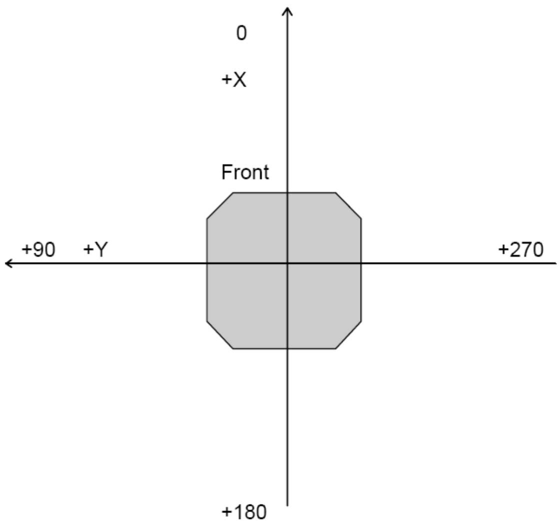


Figure 6.4 The mobile robot's heading system

The 360 degree heading is divided into four zones:

- When the heading of the mobile robot is between 0 to 45 degree and 315 to 360 degree, left wall blocks the route if the location of the goal is in quadrants 2 and 3, a right wall blocks the route if the location of the goal is in quadrants 1 and 4. In these situations the scheme suspends the ***sfAttendAtPosition*** behaviour (see Chapter 3), and resumes it when the wall is cleared.

- When the heading of the mobile robot is between 45 and 135 degree, a left wall blocks the route if the location of the goal is in quadrants 3 and 4, and a right wall blocks the route if the location of the goal is in quadrants 1 and 2. In those situations the scheme suspends the ***sfAttendAtPosition*** behaviour (see Chapter 3), and resumes it when the wall is cleared.

- When the heading of the mobile robot is between 135 and 225 degree, a left wall blocks the route if the location of the goal is in quadrants 1 and 4, and a right wall blocks the route if the location of the goal is in quadrants 2 and 3. In these situations, the scheme suspends the ***sfAttendAtPosition*** behaviour (see Chapter 3), and resumes it when the wall is cleared.

- When the heading of the mobile robot is between 225 and 315 degree, a left wall blocks the route if the location of the goal is in quadrants 1 and 2, and a right wall blocks the route if the location of the goal is in quadrants 3 and 4. In these situations the scheme suspends the ***sfAttendAtPosition*** behaviour (see chapter 3), and resumes it when the wall is cleared.

In Figure 6.3 after the robot clears the left wall, the scheme resumes ***sfAttendAtPosition*** behaviour and the robot achieves the goal.

### 6.6 The Strategy for Avoiding a Corner

A corner can be made up by a side wall and a front wall. In Figure 6.5 the goal point is (8000, 1000). There is a left hand corner in the middle of the nominal path. The robot moves into the obstruction. When it approaches the obstruction, the scheme classifies the obstruction as a left hand corner. The scheme suspends ***sfAttendAtPosition*** behaviour; robot turns once the mobile robot gets out of the corner, it resumes the global path planning and achieves the goal.

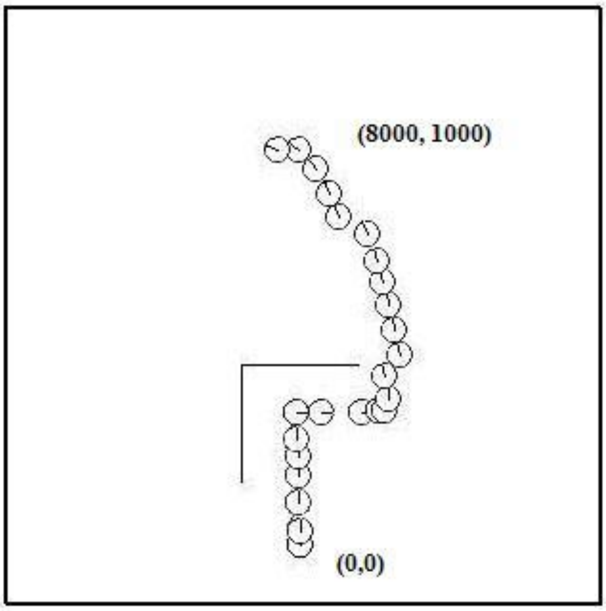


Figure 6.5, The mobile robot gets out of a left hand corner.

### 6.7 The Strategy for Avoiding a Corridor and a Dead-end

The strategy of avoiding a corridor is similar to that of avoiding a dead-end. Once the dead end is confirmed, the controller suspends the ***sfAttendAtPosition, sfConstantVelocity*** behaviours and detection activities. Then, the mobile robot continues turning the robot 180 degree to the opposite direction, resumes ***sfConstantVelocity*** behaviour and detection activities, and the situation becomes the mobile robot recognises the dead end structure as a corridor, see Figure 6.6. When the mobile robot avoids a corridor, controller resumes ***sfConstantVlocity*** that continuous moving the mobile robot. After the mobile robot gets out of the corridor the controller resumes all the detection activities and behaviours. See the example of the robot avoiding the dead-end and corridor structures, see figure the robot attends the goal point (1000, -3500).

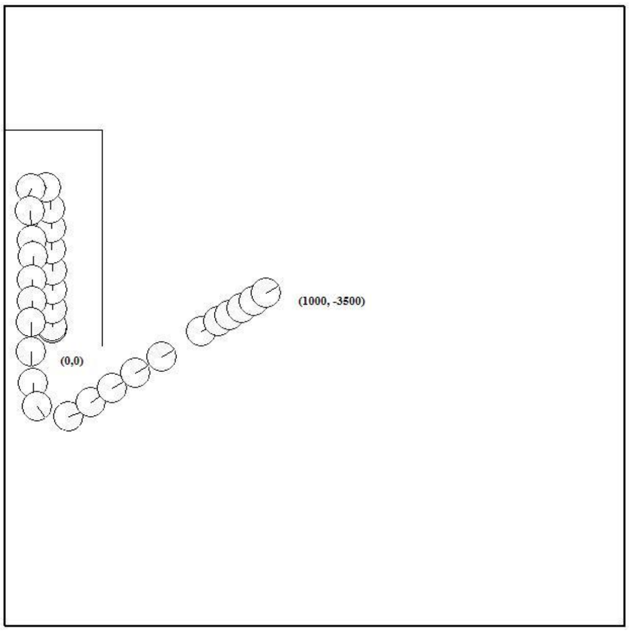


Figure 6.6 The robot gets out of a dead-end and a corridor.

### 6.8 Test and Validation of Rules

In Chapter 5 the experiments tested and verified the algorithm for detecting local structures. This section shows the limitations and some special situations in the detection, such as (a) when the robot meets multiple obstacles and the system cannot construct these obstacles as a structure within the rules; (b) when the robot meets inclined surfaces including an inclined front wall and inclined side walls; (c) when the robot meets an acute angled and an obtuse angled corner; etc. And the avoiding strategies for these situations are introduced.

#### 6.8.1 The Experiment of Objects Detection

The algorithm developed in this thesis makes use of information from sonar sensors in a manner such that a result of a decision as to whether there is a surface or an obstacle a head, etc can be made. Consider the situation where the mobile robot meets two obstacles, see Figure 6.7, one is detected by S1 and S2; the other is detected by S5 and S6. In this situation, our scheme cannot construct any possible structure of the world; as long as S3 and S4 show the status in the open range the mobile robot moves forward.

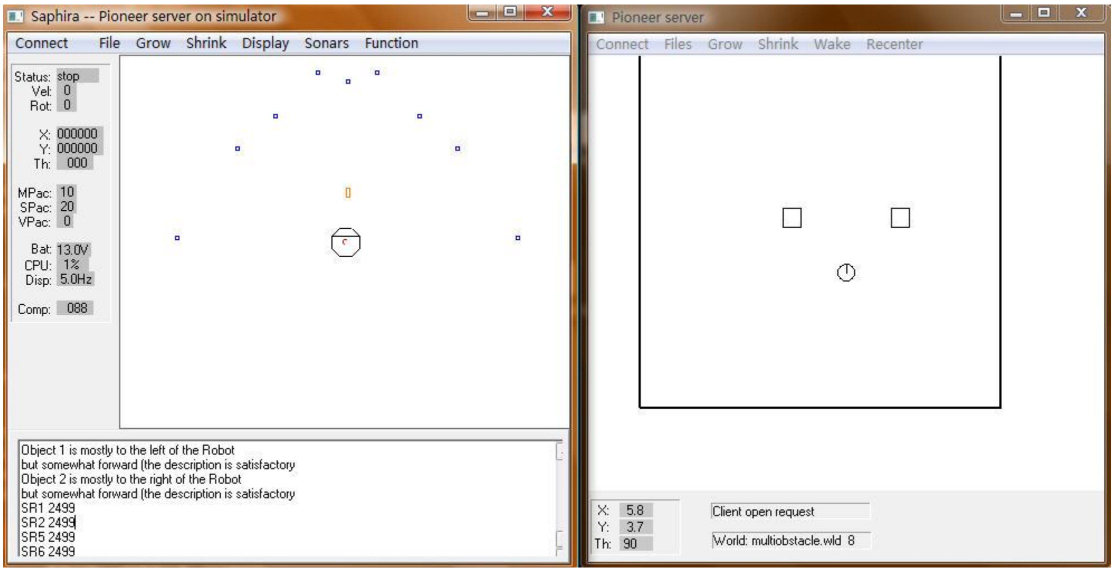


Figure 6.7 The mobile robot meeting two obstacles.

In the situation shown in Figure 6.7, the distance between two adjacent sensors are shorter than the robot diameter. Chronis' approach will detect two obstacles: Object 1 is mostly to the left of the robot, but somewhat forward. Object 2 is mostly to the right of the robot, but somewhat forward (see Figure 6.7). During the experiments, the sonar reading is showing 2499 (see Figure 6.7) which is not the actual distance between the robot and the obstacles, but is a result of the sonar specular return. The approach in this thesis is to use the only status of the sonar sensor to determine the structure. In the situation of sonar sensor failures, the fail sensor does not provide any information at all, will cause the false decision and cause collision. The Chronis' approach is not concerned with the situation with sensor failures, and bears no fault tolerance features.

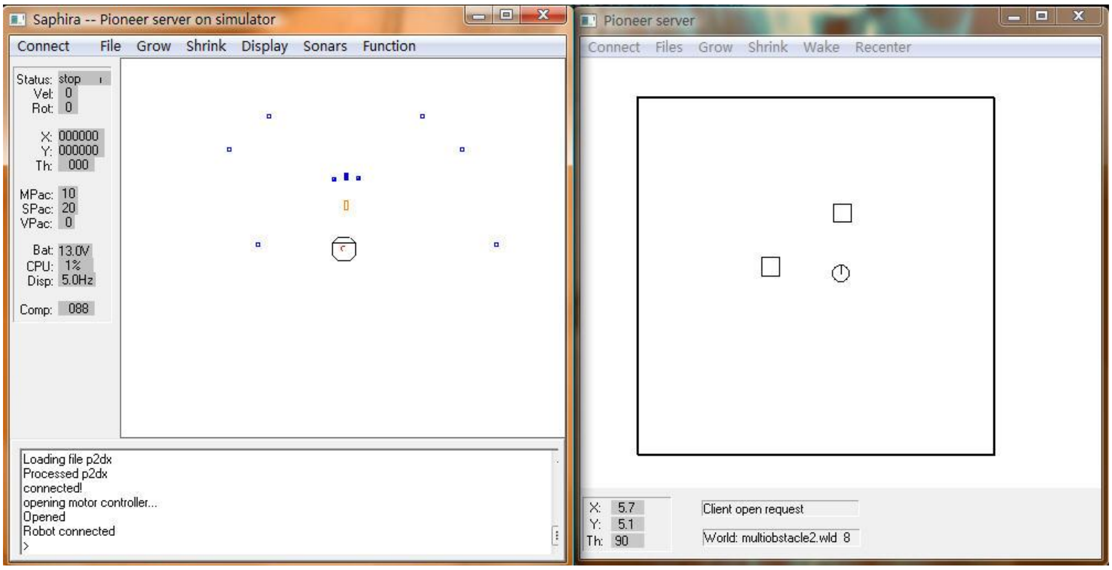


Figure 6.8 The mobile robot meeting two obstacles with front obstruction.

Figure 6.8 shows the situation when the mobile robot meets an obstruction, sensors S0, S3 and S4 are returning values. The rules cannot construct a structure for this situation, and there is no avoiding strategy for this particular situation. There are many possible situations when a structure cannot be constructed. The rules treat those “unstructured” as multi-obstacles and S3 and S4 are critical for dealing with such situations. S3 and S4 indicate necessary avoidance procedures. The avoiding strategy for multi-obstacles is the same as that of avoiding obstacles; see Figure 6.2.

### 6.8.2 The Experiment of Wall Detection

A wall is similar to an obstruction, which includes three types: a front wall, a left wall and a right wall. Chapter 5.3.3 shows how our scheme detect a surface, compared with Chronis’s approach in the experiments, see Figures 6.9 - 6.12. In the situation shown in Figure 6.9, the front wall is detected by six sonar sensors (S1-S6), which could be considered as 5 obstacles. In this case, Chronis’ method considers it as five individual obstacles, and the distance between the S3 and S4 is shorter than the diameter of the robot so the detection by S3 and S4 can be regarded as an obstacle detected by two sensor returns. The Chronis’ approach detects the rest of the 4 obstacles by S1, S2 and S5, S6 respectively. It moves mobile robot through the gap between the obstructions and will cause collision, see Chapter 7. The approach in this thesis will classify this situation as a front wall sees Chapter 5.

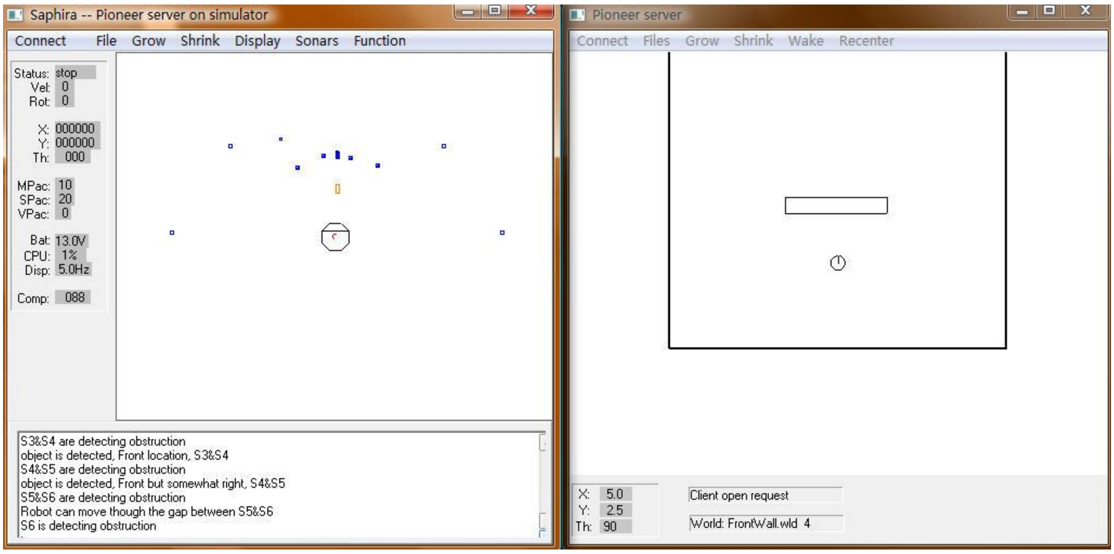


Figure 6.9 The Chronis' approach detecting the obstruction as several objects located front.

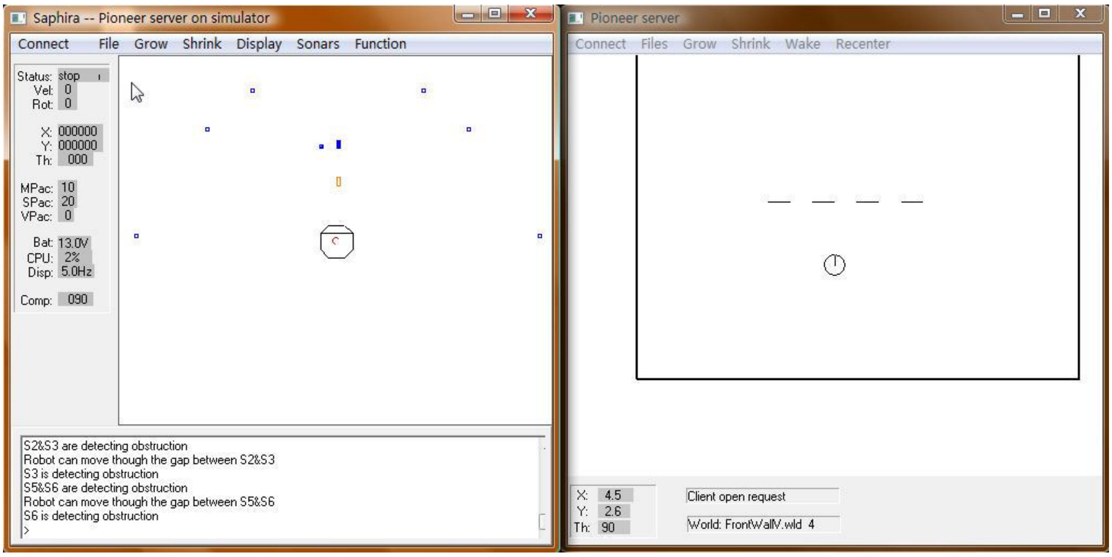


Figure 6.10 The mobile robot meeting a cluttered obstruction.

As to the situations of Figures 6.11 and 6.12, the approach taken in this thesis classifies them as a left wall and a virtual left wall respectively. Chronis' approach detects three obstacles on the left of the robot in the situation shown Figure 6.11. In the situation shown in Figure 6.12 Chronis' approach detects the situation as: *one obstacle which is somewhat forward and there are gaps between the three obstacles that can let the robot go through*.

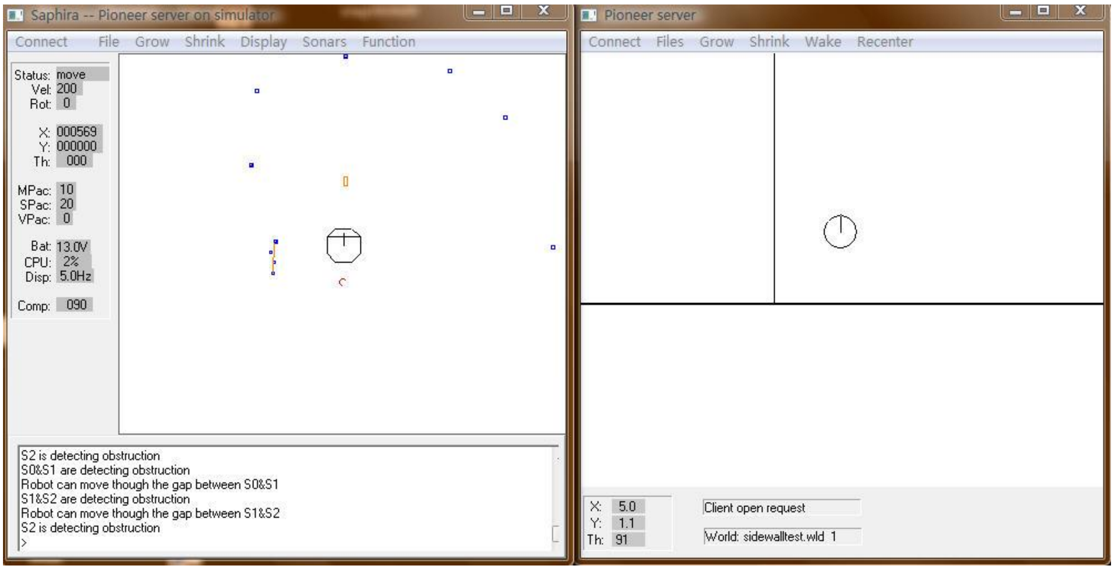


Figure 6.11 The mobile robot meets a side obstruction.

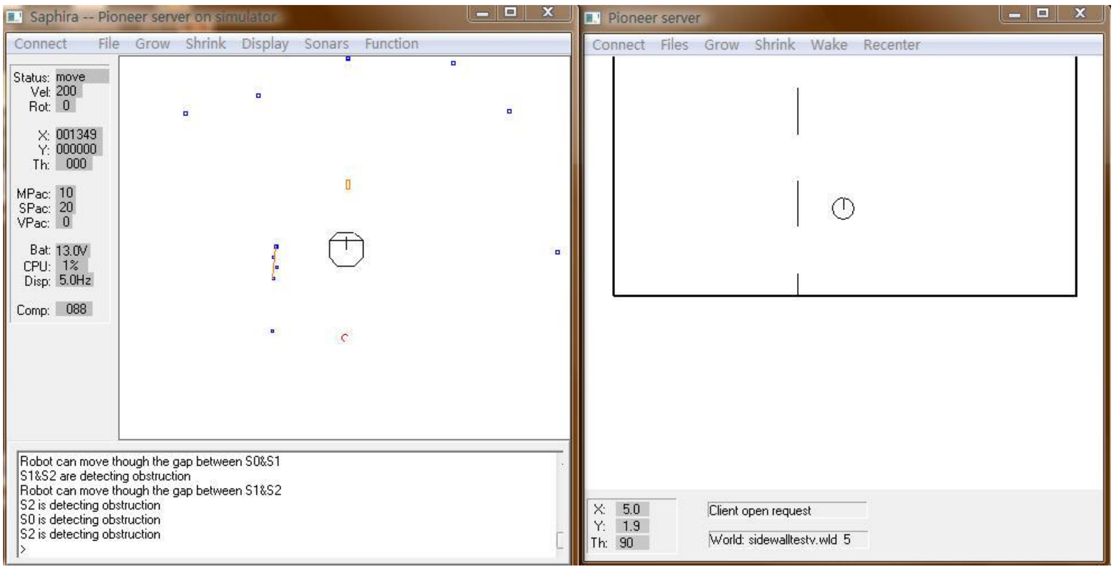


Figure 6.12 The mobile robot meeting a cluttered side obstruction.

Next, consider an inclined surface. In Figure 6.13, the scheme classifies an inclined obstruction as a front wall (see Figure 6.13). The front wall is detected by S1, S2, S3 and

S4, as long as the front obstruction is detected by S0 or S7, whatever the incline and shape, the obstruction is classified as a front wall. In detecting a front wall, it allows up to three of six sonar sensors fail at the same time. According to the restriction on designing fault tolerance rules, the critical S3 and S4 cannot fail at the same time; no adjacent sonar sensors can fail at the same time. The possible detection situations can be:

- S1, S3 and S5 are failure, S2, S4, S6 carry on for front detection.
- S1, S3 and S6 are failure, S2, S4, S5 carry on for front detection.
- S1, S4 and S6 are failure, S2, S3, S5 carry on for front detection.
- S2, S4 and S6 are failure, S1, S3, S5 carry on for front detection.

The avoidance strategy remains the same; the main front obstruction measurement is taken by S3 if S4 fails, and vice versa. When the sonar sensors fail in a cluttered environment, the structure classification is less accurate and this might cause a reduction in the ability of controlling a collision.

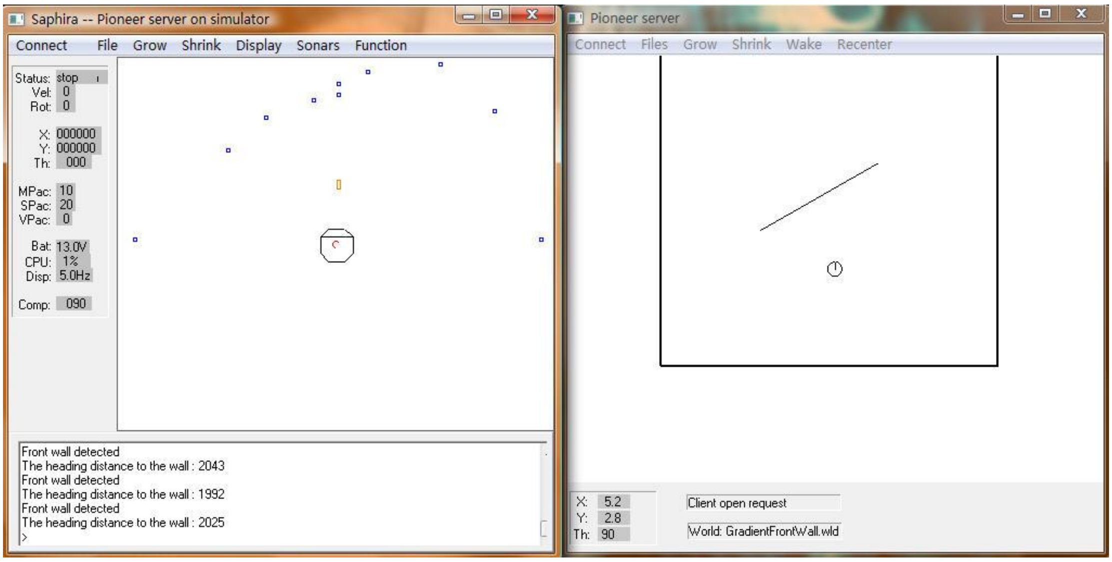


Figure 6.13 The mobile robot meeting a front inclined plane.

In Figure 6.3, it shows the trajectory of the robot avoiding the front wall; the goal point is (8000, -1000) in the Figure 6.3 which is behind the front wall. Initially the robot moves directly towards the goal point. As the robot comes close to the front wall, the robot classifies the obstruction as a front wall. The scheme suspends the behavior for attending the goal point. At the same time, the quadrant approach compares current location with the goal point, and makes a decision to instruct the robot to avoid the front wall from the right side. Once the robot clears the front wall, the scheme resumes the attending goal behavior to achieve the goal. Figure 6.14 shows the same concept of the robot avoiding an inclined front wall from the left hand side (whose goal point is (8000, 1000)). When the robot moves close to the obstruction, S1, S2 and S3 are engaged so that the robot classifies the situation as a front wall as long as S0 is in the open range status. The scheme suspends the attending goal behavior, and the quadrant system compares the location with the goal point and decides to let the robot avoid the front wall from the left.

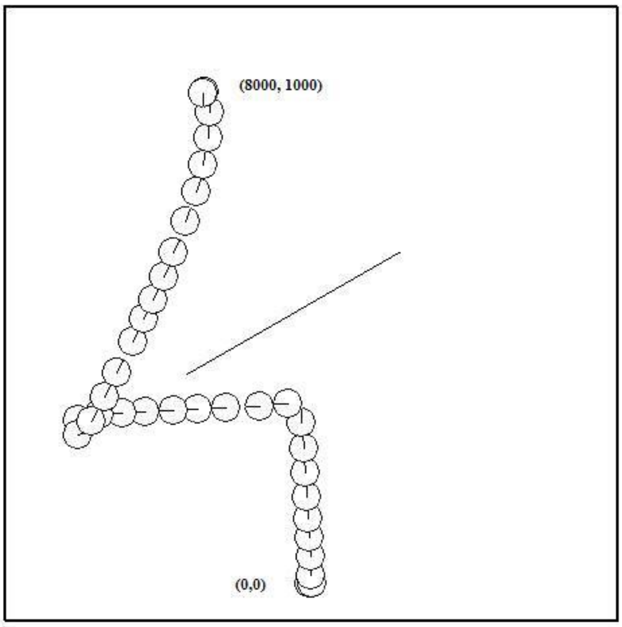


Figure 6.14 The mobile robot avoiding an inclined front wall.

The similar simulation as detecting an inclined front wall, the mobile robot classifies the situation (shown in the Figures 6.15 and 6.16) as left or right inclined plane. As long as the inclined wall does not block the critical front detection S3 and S4, the left or right wall in Figures 6.15 and 6.16 can be classified as left and right wall. A left plane can be detected by S0 and S2 when S1 fails; and when S2 fails, it can be detected by S0 and S1.

A right plane can be detected by S7 and S5 when S6 fails; and when S5 fails, it can be detected by S6 and S7. The critical S0 and S7 are required by both schemes of detection avoidance. The sufficient condition for confirming a plane is to have at least three detection sensors. The minimum condition is that it requests two detection sensors with certain geometric layout.

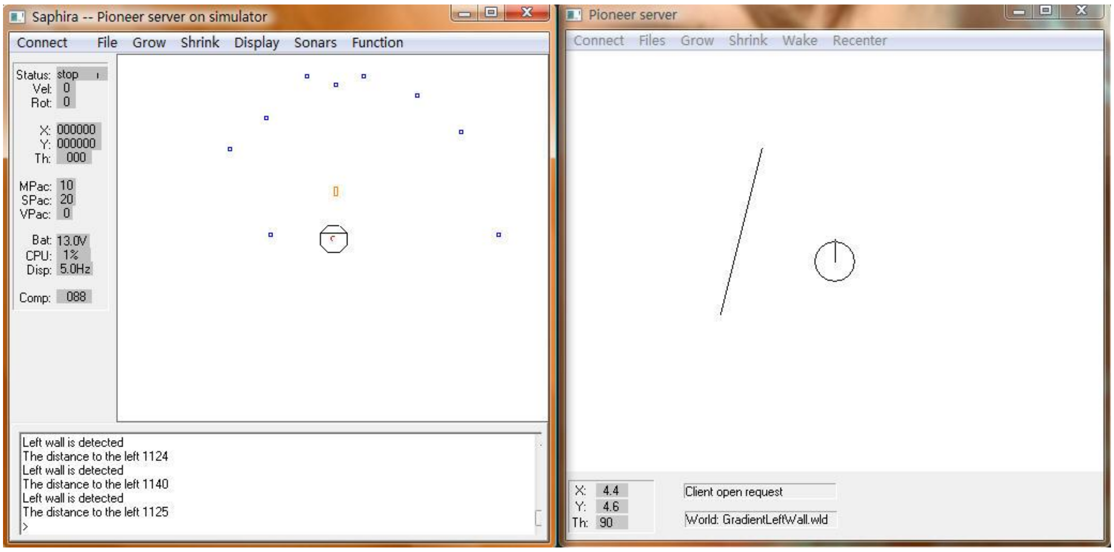


Figure 6.15 The mobile robot meeting a left inclined plane.

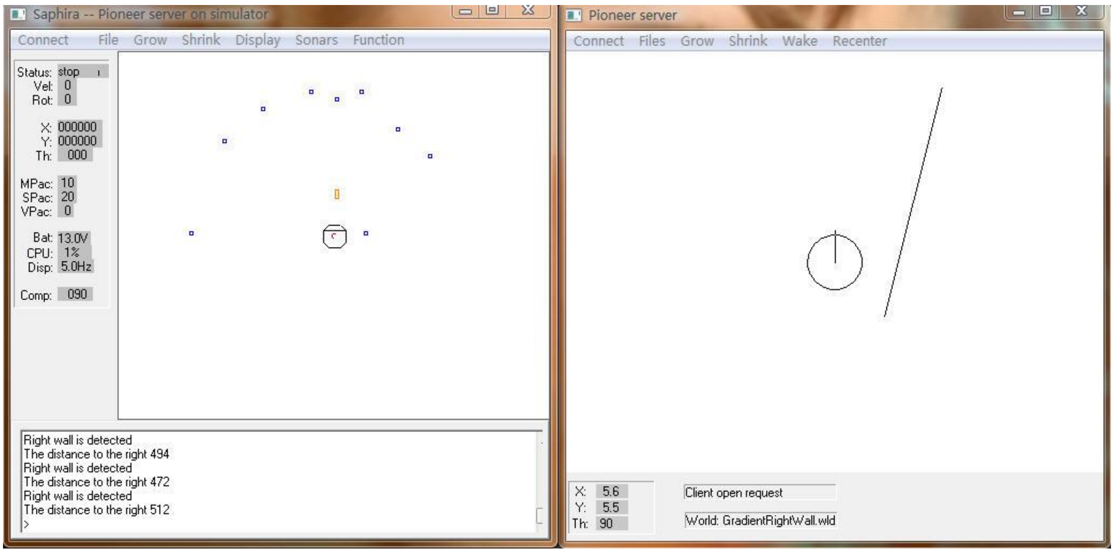


Figure 6.16 The mobile robot meeting a right inclined plane.

### 6.8.3 The Experiment of Corner Detection

Right angled corners were discussed earlier. Now consider the situation of an acute angled or an obtuse angled corner. Such corners can be considered to be made of inclined walls. Figure 6.17 and 6.18 show the special situation of corner detection. In a situation when a sonar sensor fails, it has ensured that the critical S3 and S4 have not failed together; no adjacent sonar sensors are failure at the same time; either S0 or S7 is in the detection status to confirm a left or right corner, see Chapter 4. The sufficient condition for confirming a corner type structure is four detection sensors; and at least one critical sensor from both the front and the side. The minimum condition is that it requests three detection sensors and at least one critical sensor from both the front and the side. The situations in Figures 6.17 and 6.18 are unusual geometric type of a corner type structure. However the scheme classifies them as corner structures.

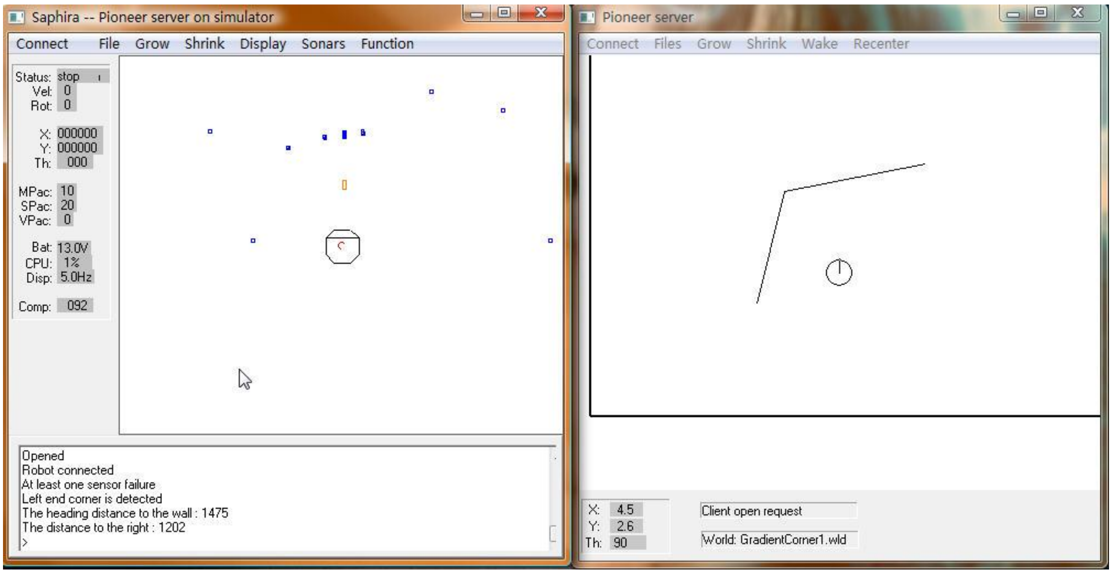


Figure 6.17 The mobile robot meeting a special left end corner.

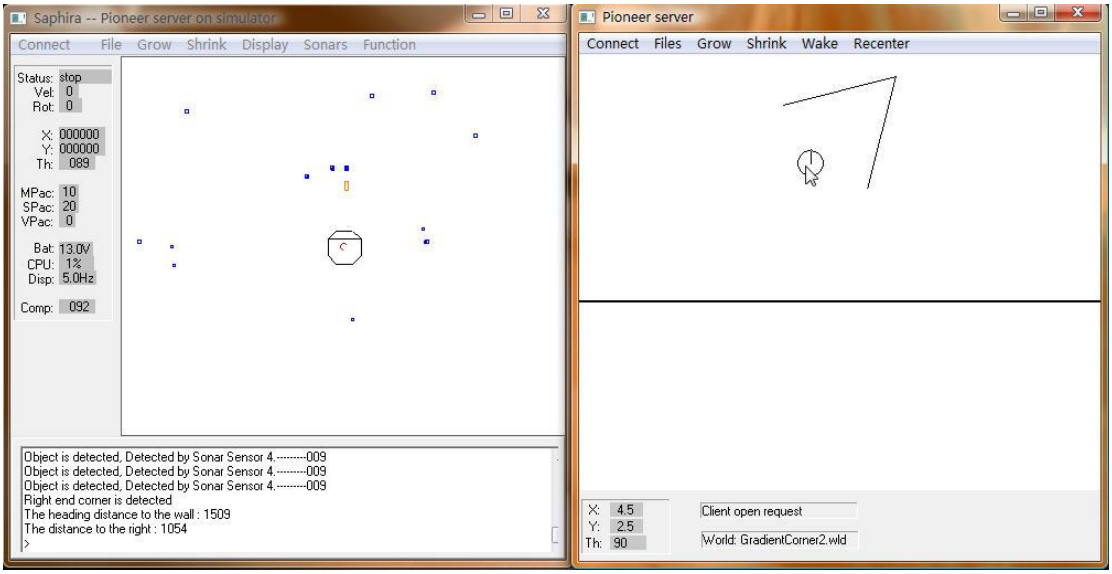


Figure 6.18 The mobile robot meeting a special right end corner.

Figure 6.5 shows how the robot can avoid the corner. When the robot is heading towards the goal point and meets the obstruction, the scheme classifies the situation as a left hand corner, and suspends the attending goal behaviour; the robot turns right, once the route is cleared, the scheme resumes the attending goal behaviour to achieve the goal.

### 6.8.4 The Experiment of Corridor and Dead-end Detection

Figure 6.19 is taken from our test programs that can show the similar connection between corridors and dead-end. The corridor is made up by a left wall and a right wall with certain distance. The important point is that S3 and S4 must keep clear at all the time when mobile robot travels in the corridor. The limitation in this thesis is that the mobile robot must travel in corridor without central obstruction, otherwise the scheme classifies the situation into dead-end structure, see Figure 6.19. However and unfortunately, the dead-end avoidance problem has a number of difficulties [Wang (2008)]. This situation may be considered in the future work. The avoidance strategy of dead end structure is turning 180 degree, transfers the dead end structure into corridor structure, and will move along with it, see Figure 6.6.

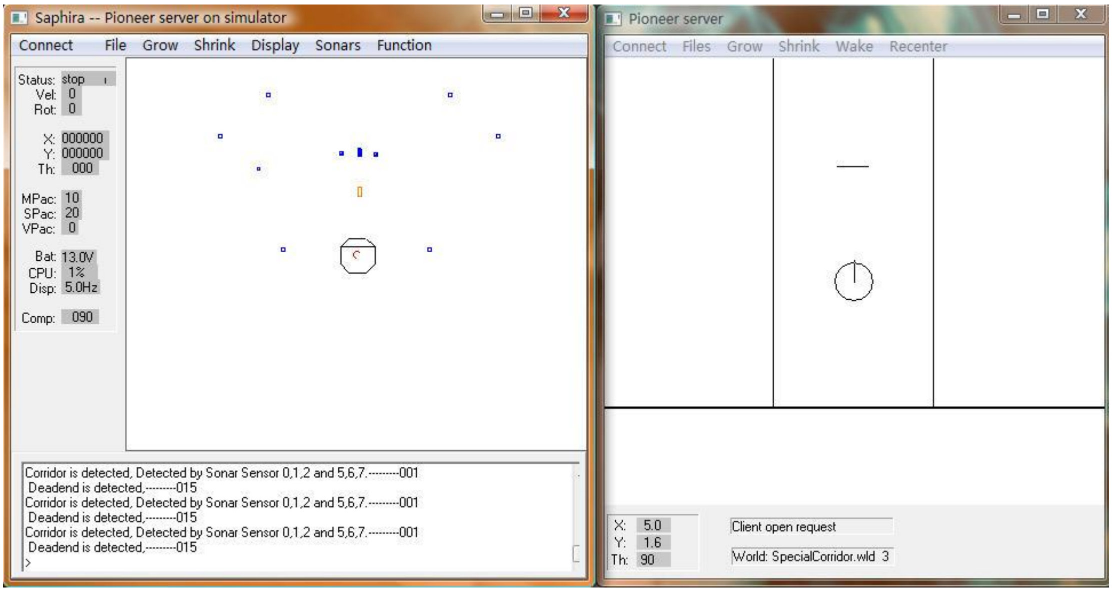

Figure 6.19 The mobile robot meets the object in the corridor.

## 6.9 Conclusion

In this chapter, a safety parameter is introduced to ensure that the robot does not collide with obstacles. In the worst case, the robot suspends all the actions to avoid collision damages to the environment and the robot itself. An intuitive quadrant system is designed in this chapter; it decides the robot directional movement when the structure is established. In the avoidance stage, each type of structure has individual avoidance strategies, the intuitive quadrant system generates the avoiding direction and the system manages the behaviours and the activities.

# Chapter 7 Path Planning in a Cluttered Environment

## 7.1 Introduction

In previous chapters, a new set of strategies for avoiding obstacles and path planning were developed (see Chapter 3-6). The strategies essentially build local maps by using the knowledge of structures and information obtained by sonar sensors. In this chapter, these strategies are put together in order to test the ability of the robot to explore an unknown and cluttered environment. Compared with Chronis' approach [2002, 2007], the approach here not only provides a unique view to identify obstacles as specific structures in a cluttered environment (see Chapter 5), but also provides strategies to avoid these structures (see Chapter 6). In addition, the scheme has an element of fault tolerance incorporated (see more details in Chapter 4). In the traditional approach global path planning in an unknown environment is designed for the robot to move around the obstacles when being blocked, for instance, using potentials and fields [Kim (2009)], and to continues when the goal direction is clear. However, the work described in Chapter 6 uses the quadrant approach with structure avoiding strategies to avoid the structure(s) from the nearer side of the goal point. In this chapter, the experiments are taken in three different environments.

## 7.2 The Experiment

In this thesis, the structures classification rules are implemented and verified in Chapter 5. In addition, Chronis' approach was also introduced and compared. In this thesis, merits mainly exist in the classification of structures especially when encountering sonar specular returns and cluttered environments. In Chapter 6, the avoidance strategies of each structure are introduced and tested. Moreover, some special situations are also analysed. In Chapters 5 and 6, the experiments for validating rules and strategies were taken as individual parts. In this chapter, more experiments will be done in three different and more complex environments with multi goal points.

### 7.3 Experiment Environment 1:

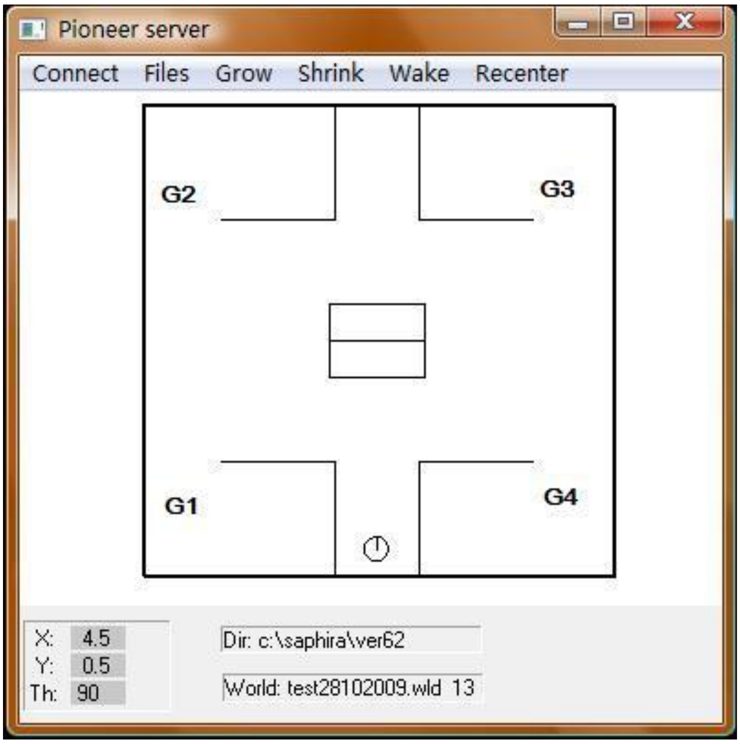


Figure 7.1 Environment 1

The experiment environment has four rooms at four corners. And there are two tables in the centre of the environment. There are four goal points, one in each room, labelled G1, G2, G3 and G4 in Figure 7.1. The environment is a 9m X 9m square indoor environment; however it is occupied and segmented by the walls and rooms. In the first experiment, the space was found narrow for the rules. The initial setup for the sonar detection range in this thesis is 3000mm, by which the procedure treats the environment as a big structure. Thus, the approach cannot proceed with a correct decision and will not lead the robot to the goal point. However, in this particular environment the sonar detection range is set to 1500mm, by which the problem mentioned above can be solved. The robot's starting point is located at the bottom of Figure 7.1. Chronis [2002, 2007] creates an approach which gathers spatial relationship information for navigation purposes, but does not build an exact model of different kinds of environment, and cannot generate a map accordingly. For example, in the case of a robot in Figure 7.1, it is at the starting point and treats the situation as there are a series of objects located by each side of it or there is one big object located by each side of it. However, by the approach in this thesis, the series of objects will be recognized as a corridor. Chronis' approach suggests if the robot can fit between the points of two adjacent sonar returns, the blocks should be taken as two objects. If the robot cannot fit between two returns, then the returns should be considered from the same object. The distance between the two adjacent sonar returns is required to be calculated and compared with the robot's diameter. The distance can be calculated by the following equation:

$$Distance = \sqrt{SRi^2 + SR_{i+1}{}^2 - 2SR_i SR_{i+1} \cos\theta} \qquad (7.1)$$

In the equation, $SRi$ and $SR_{i+1}$ refer to sonar sensor readings and $\theta$ is the angle between the adjacent sensors. Thus if the *Distance* is greater than the robot's diameter, Chronis' approach will take the case as a gap, and it also indicates the robot can pass through the gap. In this thesis, the sonar sensors are arranged at the Pioneer Robot at 20 degrees to each other, 40 degrees between S0 and S1, and between S6 and S7 is taken as the exception. The $\theta$ value should be 20 degrees and 40 degrees respectively for calculating the *Distance*. In case of the robot travels between the obstacles safely, the *Distance* should be big enough. Hence $\theta$ value is set 15 degrees and 30 degrees respectively.

### 7.3.1 Test 1: Reaching G1

The first run is to achieve G1. The coordinate of G1 point is (1000, 3000) which is located at 1m forward and 3m left of the robot. The robot always starts at (0, 0). In the course of the experiments, so as to acquire the trajectory of the robot, the testing program records every coordinate of the robot's movement. The trajectories are shown in Figure 7.2.

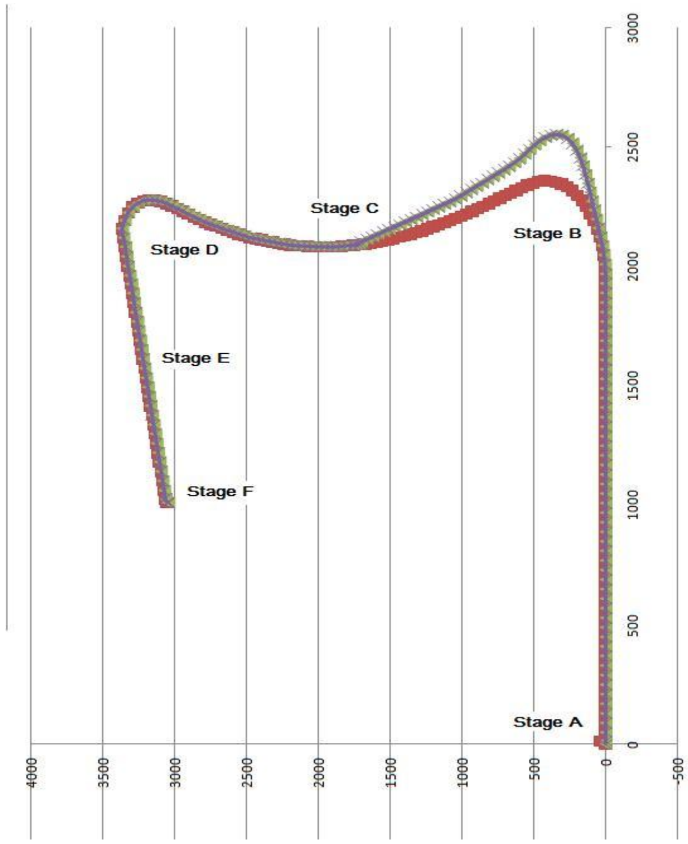


Figure 7.2 The trajectories of Environment 1 G1.

There are six stages shown in the Figure 7.2. Each stage represents the key decisions in reaching the goal of the robot. In the initial situation Stage A, the robot detects the environment as a corridor, suspends the goal attending control, and then continues moving forward. After the robot exits the corridor, which comes to Stage B, it detects a front wall. And the quadrant system compares the robot's current location with the location of the goal. It determines the goal in the third quadrant and decides to turn left. After turning left, the goal-attending behavior will lead the robot to direct to the goal point. While at the same time, the robot detects a left wall, which is Stage C, the robot suspends the goal-attending control, and moving along the wall (See Figure 7.2, the curve in Stage C is the robot moving along the left wall. The curve is caused by a component when confronting a collision, meaning that the robot is too close to the wall). In Stage D, the robot detects a front wall, and the quadrant system indicates a left turn decision. After the left turn, the robot detects a right wall. In Stage E, the quadrant system indicates that the right wall cause an obstruction to the goal point. The goal attending behavior leads the robot to the goal point. In Stage F, when the robot approaches the goal in a certain range, the scheme suspends all detection control and stops the robot. In Figure 7.2, there is a deviation between two sets of trajectories. This deviation is caused by different time of the rules make a decision (decision is triggered by certain condition such as a sensor reading, etc). The lower trajectory detects the front wall right after exiting the corridor. The top trajectory has a deviation caused by a delayed decision making. After the robot exits the corridor, sensors S0 and S7 are in the open range; the robot should detect the front wall. The delayed decision can be caused by the sonar specular return of S3 or S4. Compared with Chronis' work, at the starting point, Stage A, the robot detects three objects located at each side of the robot. The robot cannot direct access goal point; S3 and S4 are in the open range, and the robot is moving forward. In the process, the key decision is made when the robot exits the corridor, which is Stage B. His approach treats the case as three objects located in front of the robot. And the objects do not cause any obstruction to the goal point. Once S0 is in the open range, the goal attending unit leads the robot towards to the goal point.

#### 7.3.1.1 Statistical Analysis

In this experiment the total 10 goals; the successful achievement rate is 100%. And there 40% of trajectories with a "delay decision making" are in the top set and 60% of trajectories are in the lower set, see Stage B, Figure 7.2. All the stopping points are within 50 mm of goal points. In Stage B, the maximum deviation of X direction at Y = -342 is approximately 144mm.

The measure of performance of the algorithm is the mean of correctly matched decisions per goal point over the number of extracted decisions. Defining a measure by which to judge the performance of the algorithm is a very challenging task. The number of successful runs appears to be a simple solution; however, the robot may reach the destination without correctly matching all the decisions, as shown by some execution simulations and discussion later in detail in this section. The number of correctly matched decisions is not sufficient, since the algorithm would appear to perform better when more decisions are extracted and consequently matched. Chronis [Chronis, 2007] suggests defining the measure of performance of the algorithm to be P, for instance,

$$P = \sum_{j=1}^{C} \frac{R_j}{C} \tag{7.2}$$

where C is the number of total experiments per Goal point (In this thesis C= 10). And

$$R_j = 2\sum_{i=1}^{m} \frac{m-i+1}{m(m+1)} S_i \tag{7.3}$$

where *m* is the number of extracted decisions for per goal point and $S_i$ is 1 if decision *i* was matched correctly or 0 if decision *i* was not matched correctly (or not matched at all).

Table 7.1 shows the results of achieving G1. The first column indicates two approaches compared, C represents Chronis' approach and M stands for the approach utilized in this thesis. The second column shows the actual decision generated by the robot over other possible decisions. As for the robot, starting from the starting point to G1, there are six possible decisions that can be generated: Move Forward, Turn Left, Move Forward, Turn Left, Attend Goal, Stop. The third column exhibits the number of correct decisions made by each run. The fourth column displays the number of successful runs over the total possible ones. The fifth column presents performance of the algorithm. The last column shows the standard deviation from the mean of correctly matched decisions and is used to evaluate the consistency of the algorithm.

| | Generated Decisions / Total Decisions | Correct Generated Decisions | Successful runs | Performance | Standard Deviation |
|---|---|---|---|---|---|
| M | 6/6 | 6,6,6,6,6,6,6,6,6,6 | 10/10 | 1.0 | 0 |
| C | 6/6 | 6,6,6,6,6,6,6,6,6,6 | 10/10 | 1.0 | 0 |

Table 7.1 The results of environment 1 G1.

The above table represents both approaches have 10 successful runs and corrected decisions respectively.

Table 7.2 below shows the performance of structures detection. We use the same concept to measure the performance of structures detection. See the formula shown above, where *m* is the number of the extracted structures for per goal point.

| Structure Detected/Total Possible Structures | Correct Detected Structures | Performance of structure detection | Standard Deviation | Successful runs |
|---|---|---|---|---|
| 5/5 | 5,5,5,5,5,5,5,5,5,5 | 1.0 | 0 | 10/10 |

Table 7.2 The performance of structures detection for Environment 1 G1.

During the 10 successful runs, it detects all the structures that are supposed to, even in the experiment with sensor failure, see Table 7.3.

| Structure Detected with Sensor failure/Total Possible Structures | Correct Detected Structures | Performance of structure detection | Standard Deviation | Successful runs |
|---|---|---|---|---|
| 5/5 | 5,5,5,5,5,5,5,5,5,5 | 1.0 | 0 | 10/10 |

Table 7.3 The performance of structure detected with sensor failure for environment 1 G1.

### 7.3.2 Test 2: Reaching G2

The coordinate of G2 is (7000, 3000) which is located at 7m forward and 3m left of the robot. The robot always starts at (0, 0). During the experiments, the testing program recorded every coordinate of the robot movement; the results are shown in Figure 7.3.

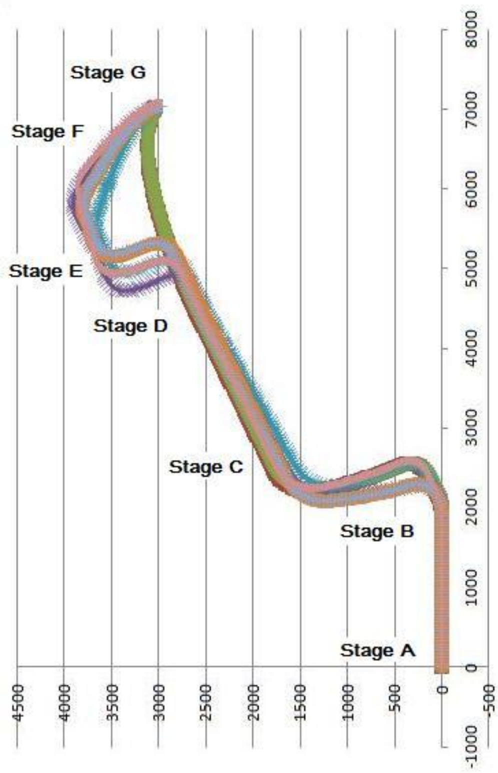


Figure 7.3 The trajectories for reaching G2 in Environment 1.

It can be seen that Stages A and B are similar to the previous goal point i.e. G1. The reason for this is that the robot is from the same the starting point. In the process of attending Goal 1, it was a deviation which happened when the robot exit the corridor. In this experiment, the deviation happens on the same stage. In Stage C, after the robot turns to the left it recognizes the wall to the left. The quadrant system indicates that the left wall does not cause any obstruction to the goal direction, which enables the goal attending behavior to work properly. In the robot's course of travelling toward the goal point, there are two possible trajectories. As can be seen from Stage G in the figure, one trajectory is the robot can directly reach the goal point without any structure being detected, which happens when the robot's crucial sensor S4 is in the open range. It leads no obstruction to the robot, and the robot moves directly to the goal point. The second trajectory shows in Stage D indicates the robot detected a front wall structure and the quadrant system decides to turn left. In Stage D, there are deviations, which are caused by the different timing for the robot to detect a front wall structure. The key point is when S4 is in detection and the robot is moving to the goal point. After the robot turns left, which is Stage E, the robot detects another front wall immediately and the quadrant system decides to turn right. After the robot's turning right, Stage F, the robot detects a left wall. It does not cause any obstruction to the goal point since the robot moves to the goal point directly. In Stage G, the robot reaches the goal point and stops. Compared with Chronis' work, in the starting point the robot detects three objects located at each side of it. The robot cannot direct access to the goal point. S3 and S4 are in the open range, and the robot moves forward. In Stage B, the robot exits the corridor, by the Chronis' approach, the result is that the obstruction is treated as three individual objects (see Figure 7.3). The indication is the robot can move through the gap between S2 and S3. The quadrant system leads the robot to the direction between S2 and S3. After the robot's turning, the Chronis' approach detects the objects by S3 and S4, the quadrant system decides to turn left. In Stage C, it detects the objects, though they do not make any obstruction to the goal point, and the robot moves to the goal point. As mentioned, there are two possible ways in Stage C. One is the robot can directly reach the goal point. The other way is triggered when S4 is in detection and two objects are detected by S4 and S5. The indication is a gap, through which the robot can go. This wrong decision will lead the quadrant system to let the robot go through the gap which does not exist at all.

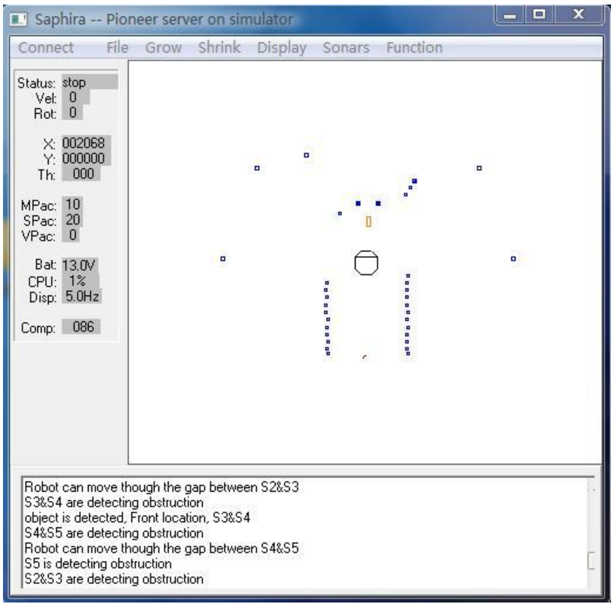


Figure 7.4 The specular return leading to a wrong decision

#### 7.3.2.1 Statistical Analysis

In this experiment, the total successful goal achievement is 100% of this approach. And there are two sets of trajectory, there are 60% of trajectories are in the top set and 40 % of trajectories are in the lower set at Stage B. The deviation of X direction at Y=342 is approximately 144mm. In Stage C, there are 60% routes detected around the structure and 40% routes directly going to the goal point. In Stage D, there are 50% deviation trajectories; and there are 50% trajectories are in the expected set of trajectories which are the top set in Stage D. The deviation of X direction is approximately 172 mm at Y=3500. All the stopping points are within 50 mm of the goal points. In Chronis' approach, there are 40% of runs are successful.

Table 7.4 shows the result of achieving G2. The columns are the same as the previous table. The second column shows the actual decision made by robot over the rest of the possible decisions. From the starting point to G2, there are seven possible decisions can be generated for the robot: Move Forward, Turn Left, Attend Goal, Turn Left, Turn Right, Attend Goal, and Stop.

| | Generated Decisions / Total Decisions | Correct Generated Decisions | Successful runs | Performance | Standard Deviation |
|---|---|---|---|---|---|
| M | 7/7 | 7,7,4,7,4,7,4,7,7,4 | 10/10 | 0.91 | 1.55 |
| C | 5/7 | 2,2,3,2,3,2,3,2,2,3 | 4/10 | 0.68 | 0.52 |

Table 7.4 The results of reaching G2 in Environment 1.

As shown by the table, the approach put forward here had 10 successful runs and six of them made respective decisions. In this particular case, the robot can achieve the goal point without detecting all the structures about and following different routes which might divert by different decisions, hence there are a large number of standard deviations. By Chronis' work, there are 4 successful runs.

The table below shows the performance of structures detection.

| Structure Detected/Total Possible Structures | Correct Detected Structures | Performance of structure detection | Standard Deviation | Successful runs |
|---|---|---|---|---|
| 5/5 | 5,5,3,5,3,5,3,5,5,3 | 0.92 | 1.03 | 10/10 |

Table 7.5 The performance of structures detection for Environment 1 in reaching G2.

During the 10 successful runs, it detects all the structures that are supposed to be detected.

The table below shows the performance of the structures detected with sensors failure. In the process of detection the approach proposed in thesis converts the condition of sensors failure into detection, which makes more votes for detecting structures, see details in Chapter 4. In this experiment, all the structures are detected even with sensors failure.

| Structure Detected with Sensor failure/Total Possible Structures | Correct Detected Structures | Performance of structure detection | Standard Deviation | Successful runs |
|---|---|---|---|---|
| 5/5 | 5,5,5,5,5,5,5,5,5,5 | 1.0 | 0 | 10/10 |

Table 7.6 The performance of structure detected with sensor failure for environment 1 G2.

### 7.3.3 Test 3: Reaching G3

The coordinate of G3 is (7000, -3000) which is located at 7m forward and 3m right of the robot. The robot always starts at (0, 0). During the experiment, to achieve the trajectory of the robot, the testing program records every coordinate of the robot movement and the results are shown in the Figure 7.5. Since G3 is a reflection of G2, it is expected that the results will be mirrored in this experiment. However, Figure 7.5, it can be seen, the trajectories, apart from Stages A andB, are not reflected from previous experiment.

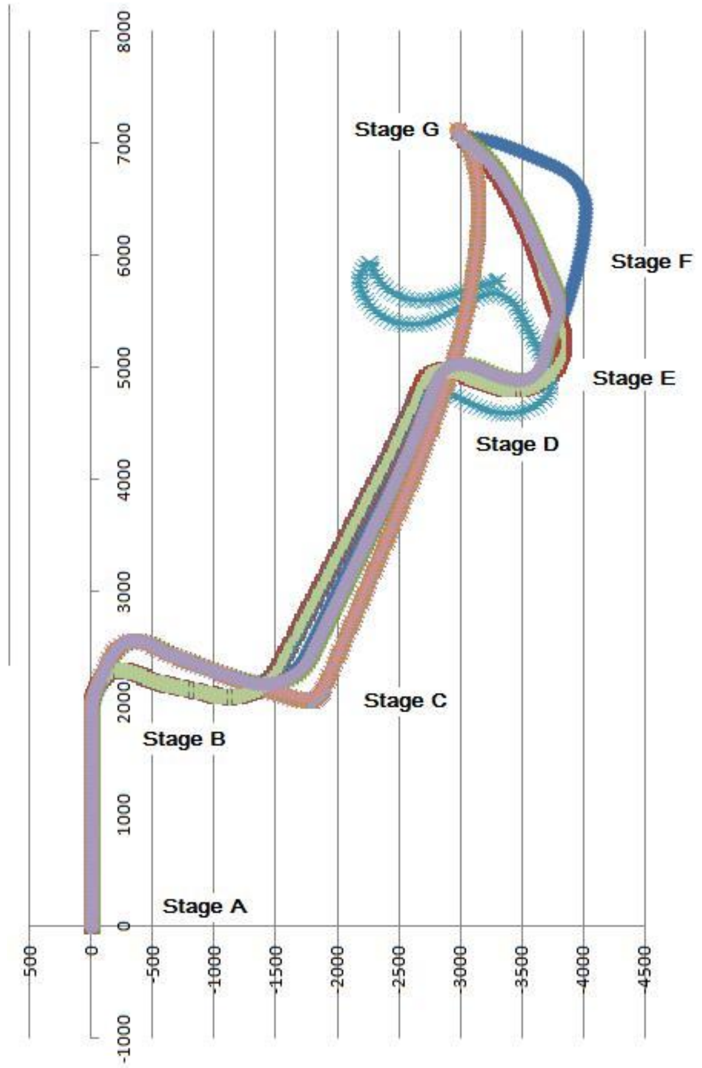


Figure 7.5 The trajectories for reaching G3 in Environment 1.

In the process of attending Goal 1, 2, it is a deviation which happened when the robot exits the corridor. In this experiment, the deviation happens on the same stage, see Figure 7.5. In Stage C, after the robot turns to the right, it recognizes the wall located in the right. And the quadrant system indicates that the right wall does not cause any obstruction in the direction of goal, which enables the goal attending behavior to work. In the robot's travel toward the goal point, there are two possible trajectories. By Stage G in the figure, one trajectory can be seen, where the robot can directly reach the goal point without any structure being detected. The case is when the robot's crucial sensor S3 is in the open range, and the robot is led to move directly to the goal point. The second trajectory in Stage D shows that the robot detects a front wall structure and the quadrant system decides to turn right. In stage D, there are deviations which might be caused according to the different timing for the robot to detect a front wall structure. The key point is when S3 is in detection and the robot is moving to the goal point. There is an experiment when failing to reach the goal point. The quadrant system decides to turn left when the robot detects the front wall in Stage D. After the robot's turning right, Stage E, the robot detects another front wall immediately and the quadrant system decides to turn left. After the robot's turning left, Stage F, the robot detects a right wall. However, it does not cause any obstruction to the goal point. The robot moves to the goal point directly. In this stage there is an experiment, where the robot delays in responding and is caused by the Sonar sepcular return. In Stage G, the robot reaches the goal point and it stops. Compared with Chronis' work, in the starting point the robot detects three objects located at each side of it. The robot cannot get direct access to the goal point; S3 and S4 are in the open range, and the robot moves forward. In Stage B, the robot exits the corridor, and the obstruction will be detected as three individual objects by the Chronis' approach, see Figure 7.4, and it also indicates that the robot can move through the gap between S4 and S5. The quadrant system leads the robot to the direction between S4 and S5. After the robot's turning, the Chronis' approach detects the objects by S3 and S4, and the quadrant system decides to turn right. In Stage C, it detects the objects, which do not make any obstruction to the goal point. The robot moves to the goal point. As mentioned above, there are two possible ways in Stage C. One is the way by which the robot can directly reach the goal point. The other is the one triggered when S3 is in detection. As long as S3 is in detection, two objects are detected by S2 and S3, and it indicates there is a gap where the robot can go through. This wrong decision will lead the quadrant system to let the robot go through the gap which does not exist.

#### 7.3.3.1 Statistical Analysis

In this experiment, the total successful goal achievement is 90% of this approach. And in Stage B, there are two sets of trajectory. There are 70% of trajectories are in the top set and 30 % of trajectories are in the lower set. The deviation of X direction at Y= -350 is approximately 138 mm. In Stage C, there are 60% routes detecting the structure and 40% routes directly moving to the goal point. The deviation of X direction at Y= -1761 is approximately 344 mm. In Stage D, there are 17% deviation trajectories; and there are 33% trajectories are in the top set in Stage D. The deviation of X direction is approximately 104 mm at Y=-3500. All the stopping points are within 50 mm of goal points. In Chronis' approach, there are 40% of runs are successful.

Table 7.7 shows the result of achieving G3. The columns are the same as the previous table. The second column shows the actual decision generated by the robot over the total possible decisions. From the starting point to G3, there are seven possible decisions made for the robot: Move Forward, Turn Right, Attend Goal, Turn Right, Turn Left, Attend Goal, and Stop.

| | Generated Decisions / Total Decisions | Correct Generated Decisions | Successful runs | Performance | Standard Deviation |
|---|---|---|---|---|---|
| M | 7/7 | 7,5,4,7,4,7,7,4,4,7 | 9/10 | 0.9 | 1.51 |
| C | 5/7 | 2,2,3,2,3,2,3,2,2,3 | 4/10 | 0.68 | 0.52 |

Table 7.7 The results of reaching G3 in Environment.

This is a situation similar to G2, in which the robot can achieve the goal point without detecting all the structures and following different routes diverted by different decisions, and there are a large number of standard deviations. Compared with Chronis' work, there are 4 successful runs.

Table 7.8 below shows the performance of structure detection.

| Structure Detected/Total Possible Structures | Correct Detected Structures | Performance of structure detection | Standard Deviation | Successful runs |
|---|---|---|---|---|
| 5/5 | 5,4,3,5,3,5,5,3,3,5 | 0.91 | 0.99 | 9/10 |

Table 7.8 The performance of structures detection in reaching G3 in Environment 1.

Table 7.9 below shows the performance of structures detected with sensors failure. The situations similar to G2, all the structures are detected with sensor failure. In this particular experiment the successful runs are 100%.

| Structure Detected with Sensor failure/Total Possible Structures | Correct Detected Structures | Performance of structure detection | Standard Deviation | Successful runs |
|---|---|---|---|---|
| 5/5 | 5,5,5,5,5,5,5,5,5,5 | 1.0 | 0 | 10/10 |

Table 7.9 The performance of structure detected with sensors failure reaching G3 in Environment 1

### 7.3.4 Test 4: Reaching G4

G4's coordinates are (1000, -3000), which is located at one meter forward and three meters right of the robot. The robot always starts at (0, 0). In the experiment, to achieve the trajectory of the robot, the testing program records every coordinate of the robot's movement. The results are shown in Figure 7.6.

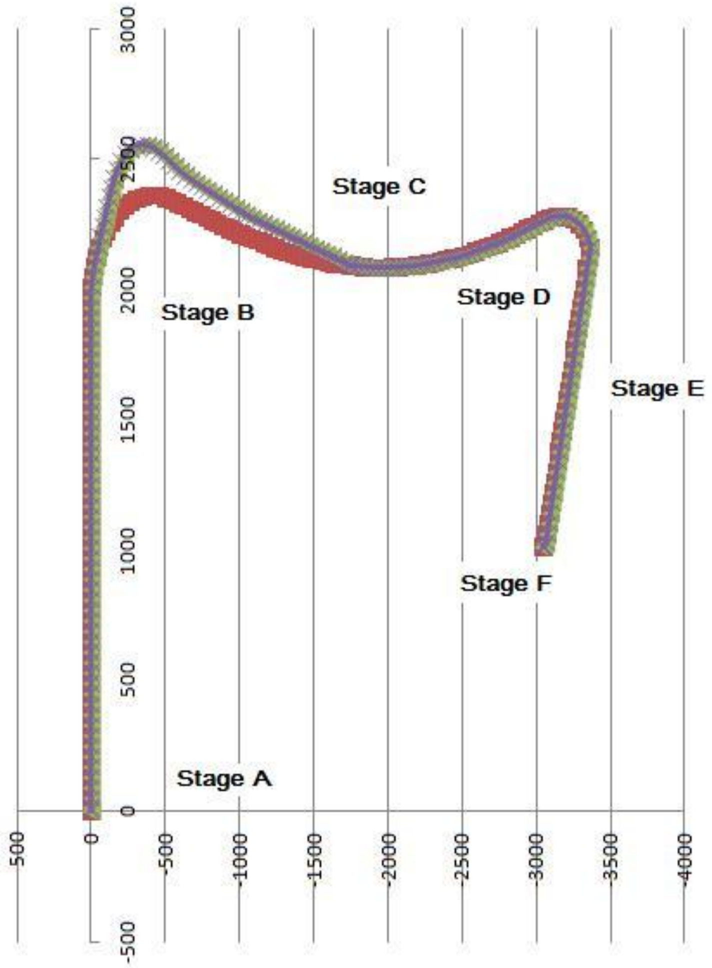


Figure 7.6 The trajectories for reaching G4 in Environment 1.

There are six stages shown in the figure. In the initial Stage A, the robot detects the environment as a corridor, suspends the goal attending control, and then keeps moving forward. After the robot exits the corridor, Stage B, it detects a front wall. The quadrant system compares the robot's current location with the location of the goal, determining the goal in the first quadrant and decides to turn right. After the turning right, the goal attending behavior will lead the robot direct to the goal point. At the same time, the robot detects a right wall. It comes to Stage C. The robot suspends the goal attending control and moves along the wall (See Figure 7.6, the curve in Stage C shows that the robot is moving along the right wall. The curve is caused by a unit in the case of a collision and the robot is too close to the wall). In Stage D, the robot detects a front wall. The quadrant system indicates a decision of making a right turn. After turning right, the robot detects a left wall. It comes to Stage E, in which the quadrant system indicates the left wall will not make an obstruction to the goal point. The goal attending behavior leads the robot to the goal point. In Stage F, when the robot approaches the goal in the certain range, the scheme suspends all detection control and stops the robot. In Figure7.6, there is a deviation between the two sets of trajectories. The trajectory in the lower set detects the front wall right after the robot exiting the corridor. The top trajectory has a deviation caused by a delayed decision. After the robot exits the corridor, S0 and S7 are in the open range; the robot should have detected the front wall. The delayed decision can be caused by the sonar specular return of S3 or S4. Compared with Chronis' work, in the starting point, Stage A, the robot detects three objects located at each side of it. The robot cannot get a direct access to the goal point; S3 and S4 are in the open range, and the robot is moving forward. In the process, the key decision is made when the robot exits the corridor. It is Stage B, Chronis' approach treats the case like this as three objects located in front of the robot. And the objects do not make any obstruction to the goal point. Once S7 is in the open range, the goal attending unit leads the robot to the goal point.

#### 7.3.4.1 Statistical Analysis

In this experiment the total successful goal achievement is 100%. And there are 60% of trajectories are in the top set and 40% of trajectories are in the lower set. All the stopping points are within 50 mm of goal points. Table 7.10 shows the results of achieving G4. The second column shows the actual decision generated by the robot over the total possible decisions. From the starting point to G4, there are six possible decisions generated for the robot: Move Forward, Turn Right, Move Forward, Turn Right, Attend Goal, and Stop.

| | Generated Decisions / Total Decisions | Correct Generated Decisions | Successful runs | Performance | Standard Deviation |
|---|---|---|---|---|---|
| M | 6/6 | 6,6,6,6,6,6,6,6,6,6 | 10/10 | 1.0 | 0 |
| C | 6/6 | 6,6,6,6,6,6,6,6,6,6 | 10/10 | 1.0 | 0 |

Table 7.10 The results of reaching G4 in Environment 1.

The table shows both approaches have 10 successful runs and make correct decisions respectively. The table below shows the performance of structures detection.

| Structure Detected/Total Possible Structures | Correct Detected Structures | Performance of structure detection | Standard Deviation | Successful runs |
|---|---|---|---|---|
| 5/5 | 5,5,5,5,5,5,5,5,5,5 | 1.0 | 0 | 10/10 |

Table 7.11 The performance of structures detection for reaching G4 in Environment 1.

During the 10 successful runs, it detects all the structures that are supposed to be detected. The table below shows the performance of structure detected with sensor failure.

| Structure Detected with Sensor failure/Total Possible Structures | Correct Detected Structures | Performance of structure detection | Standard Deviation | Successful runs |
|---|---|---|---|---|
| 5/5 | 5,5,5,5,5,5,5,5,5,5 | 1.0 | 0 | 10/10 |

Table 7.12 The performance of structure detected with sensors failure in reaching G4 in Environment 1

### 7.3.5 An Overview of Environment 1 Experimenting Results

Environment 1 is a simple structure environment which has four goal points. There are two goal points on the left side and the other two are symmetrical to the right. In Stage A all the experiments have the result of detecting a corridor structure. In Stage B, from the experiment, all the front wall structures are detected in spite of a deviation due to the timing of the structure detection and the quadrant system's decision making. At first, there is overall 41% of delay making decisions. However, in this stage, there are 100% experiments detecting correct structures in spite of sensors failure. Of all the experiments, there are 39/40 successful runs; 40/40 successful runs with sensors failure and 28/40 successful runs of Chronis' approach. The overall performance and successful rate are shown in the following table:

| | Average Performance | Successf Rate |
|---|---|---|
| M | 0.95 | 0.975 |
| C | 0.84 | 0.7 |

Table 7.13 Overall performance and success rate.

The overall structure detection performance and structure detection performance with sensor failure is shown in the following table:

| Average Structure Detection Performance | Average Structure Detection Performance with Sensor Failure |
|---|---|
| 0.96 | 1.0 |

Table 7.14 Average structure detection performance

## 7.4 Experiment Environment 2:

The simulation environment is designed to include corridors, obstacles of various shapes and sizes in order to test the performance of the various methods. These obstacles are cluttered in the environment, for example, some are in the middle, and some are along the sides (see Figure 7.7). The environment is a10m x 10m square space. The robot starts in the corridor, see Figure 7.7. Each test starts from the starting point and with G1, G2, G3, and G4 as goals respectively. The corridor can be thought of as being constructed by left and right side walls, and the junction in Figure 7.7 can be classified as a corner. The corridor has both a right turn and a left turn. The robot detects the corridor and moves forward through the corridor. At the junction, the robot detects a corner and takes a right turn at the end and then a left turn to avoid corner.

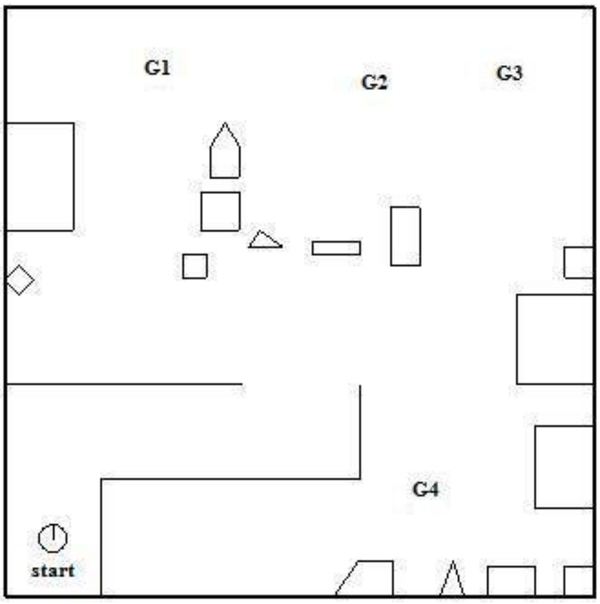


Figure 7.7 Environment 2.

In addition there are several different objects on the boundary of the environment; these objects can cause different sonar reading returns along with specular return and uncertainty in the readings. In the environment, there are six objects cluttered in the middle. They are not laid out on the same level vertically and horizontally. When the robot meets the cluttered objects, it could classify them: as a front wall when the robot is facing the lower four objects, or as a side wall when passing three objects which are laid out vertically. Those obstacles are located in different distances, and there is a problem which when two neighboring sonar sensors return how to decide whether the readings are from a single obstacle or multiple obstacles.

### 7.4.1 Test 1: Reaching G1

The Goal 1's coordinate is (8000, -2000) which is located at 8m forward and 2m right of the robot. The robot always starts at (0, 0). During the experiments, the testing program records every coordinate of the robot's movement, and the results are shown in Figure 7.8.

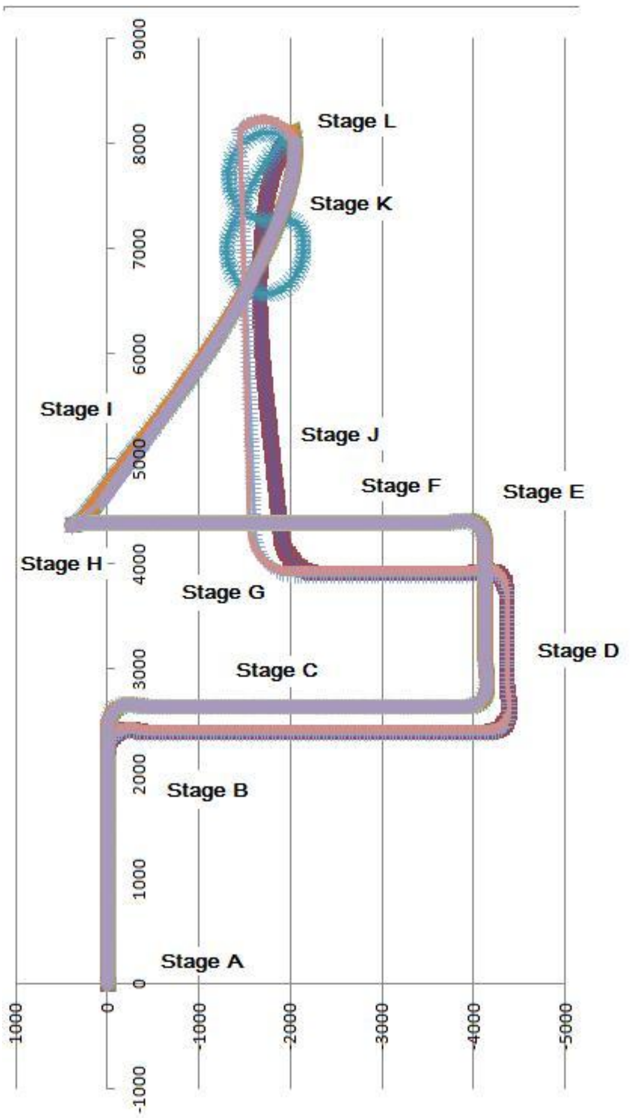


Figure 7.8 The trajectories for reaching G1 in Environment 2.

There are twelve stages shown in Figure 7.8. In the initial Stage A, the robot detects the environment as a corridor, suspends the goal attending control, and then keeps moving forward. After the robot exits the corridor, it comes to Stage B. It detects a left corner, and decides to turn right. After turning right, in Stage C, the robot detects a corridor, suspends the goal attending control, and moves forward. In Stage D, the robot detects a right corner; the system decides to turn left. After exiting the corner, in Stage E, the robot detects a front wall. As seen from the figure, there are four objects in front of the robot. The robot detects this situation possibly as a virtual front wall, which is determined by the detecting sensors and the rules. This virtual type of structure is introduced in Chapters 3 and 4. The quadrant system decides to turn left. Also there is a deviation of two set of routes, which resulted from the timing for making decisions. In Stage F, the robot detects sort of a corridor structure. The structure could be a virtual corridor. The system makes the decision for the robot to move forward. In Stages G and H, they represent two routes. In Stage G, the robot detects a left wall. The quadrant system decides to move towards the goal point. In Stage J, the robot detects a corridor structure, though it could be a virtual corridor. The quadrant system decides to move forward. After the robot exits the corridor, it comes to Stage K. The quadrant system leads robot to the direction of the goal. In Stage L, the robot approaches the goal point and stops at goal point. On the other hand, Stage H, the robot fails to turn left at Stage G and to detect structures, resulting from the deviation in Stage E. The robot detects a front wall in Stage H, and the quadrant system decides to direct the robot towards the goal point. In Stage I, the robot detects a corridor structure, though the corridor could be a virtual corridor. And the robot moves forward. After the robot exits the corridor, it comes to Stage K. The quadrant system guides the robot to the direction of the goal. In Stage L, the robot approaches the goal point and stops at the goal point. There is one experiment failing to reach the goal point. When it is in Stages K and L, there is a unit to measure whether the robot approaches the goal point less than a meter or not. When the robot is less than a meter to the goal point, the unit suspends all the activities except for the goal attending behavior. When the robot reaches the goal point, it stops. However, in this particular experiment, when the robot is in the corridor, this unit is suspended. When the robot exits the corridor, it detects another structure (which is a right corner in this experiment, S3 reading is bounced from the top boundary of the room) rather than enabling the unit to work.

#### 7.4.1.1 Chronis' Approach

In the initial situation, Figure 7.9, the instruction in the left window shows that S0, S1 and S2 are engaged. On the right hand side, S5, S6 and S7 are engaged. In this case, the robot detects that there are three objects on the left. And there are three objects located on the right. S3 and S4 are front detection sensors; they are in the open range, which enables the robot to keep moving forward.

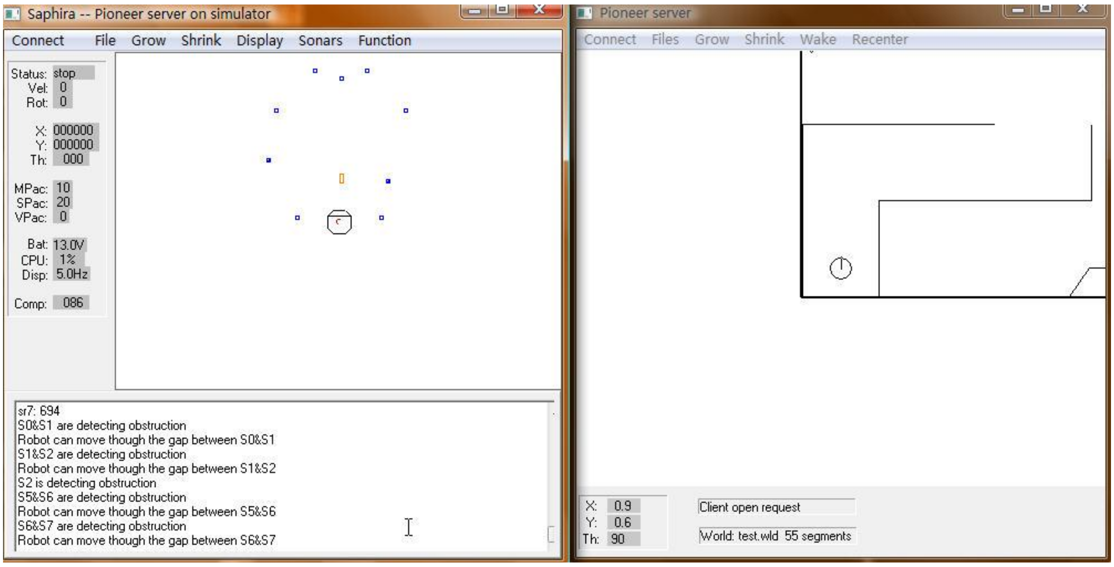


Figure 7.9 Step 1 of Chronis' approach.

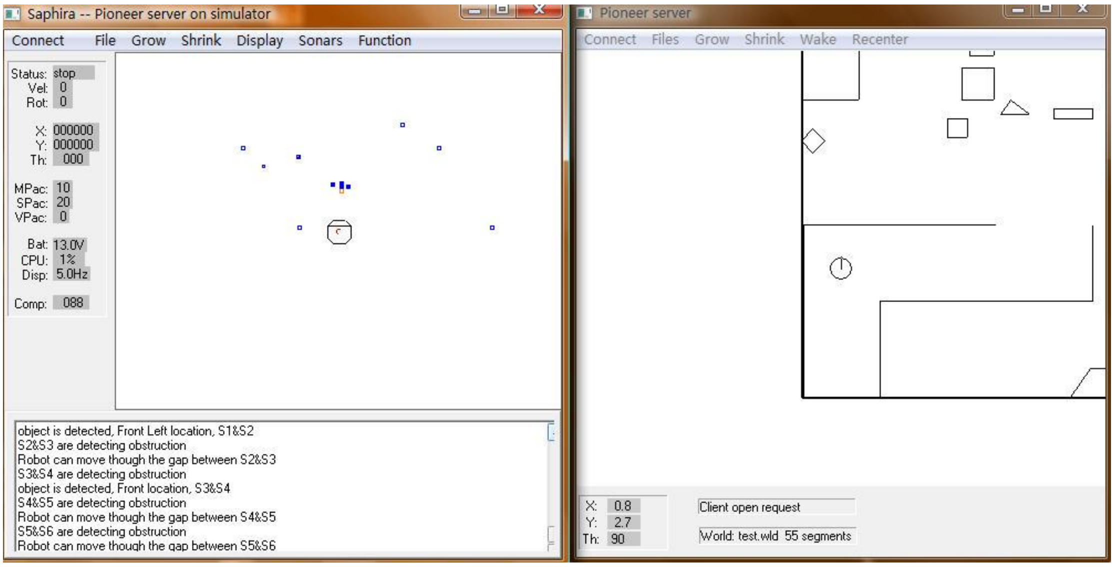


Figure 7.10 Step 2 of Chronis' approach.

In Stage B, see Figure 7.10, the left window indicates that S0, S1…S6 are all engaged, and the distance between adjacent sensors are shorter than the diameter of the robot. The quadrant system determines to turn right.

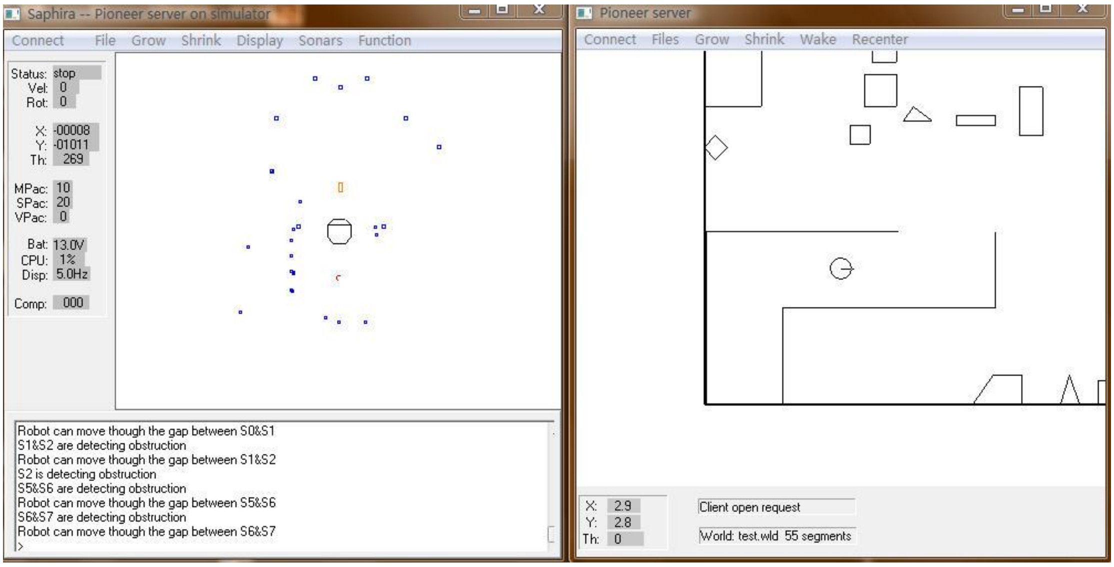


Figure 7.11 Step 3 of Chronis' approach.

The Stage C, see Figure 7.11, which is similar to the Stage A. the robot keeps moving forward.

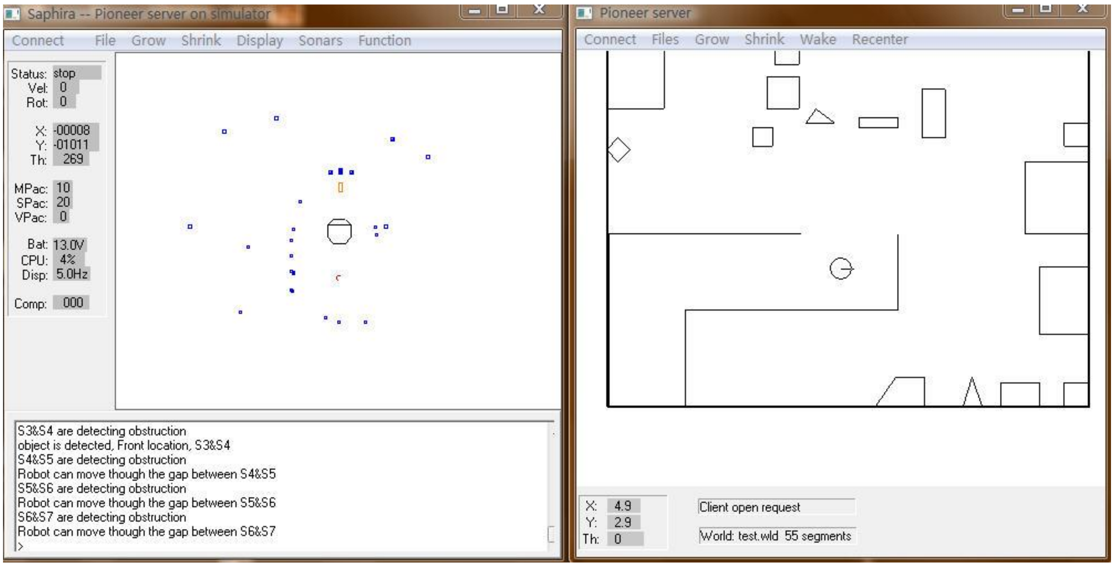


Figure 7.12 Step 4 of Chronis' approach.

In Stage D, see Figure 7.12, a case similar to Stage B, the instruction in the left window shows that S1, S2…S7 are all engaged, and the distance between adjacent sensors are shorter than the diameter of the robot. The quadrant system makes the decision that the robot turns left.

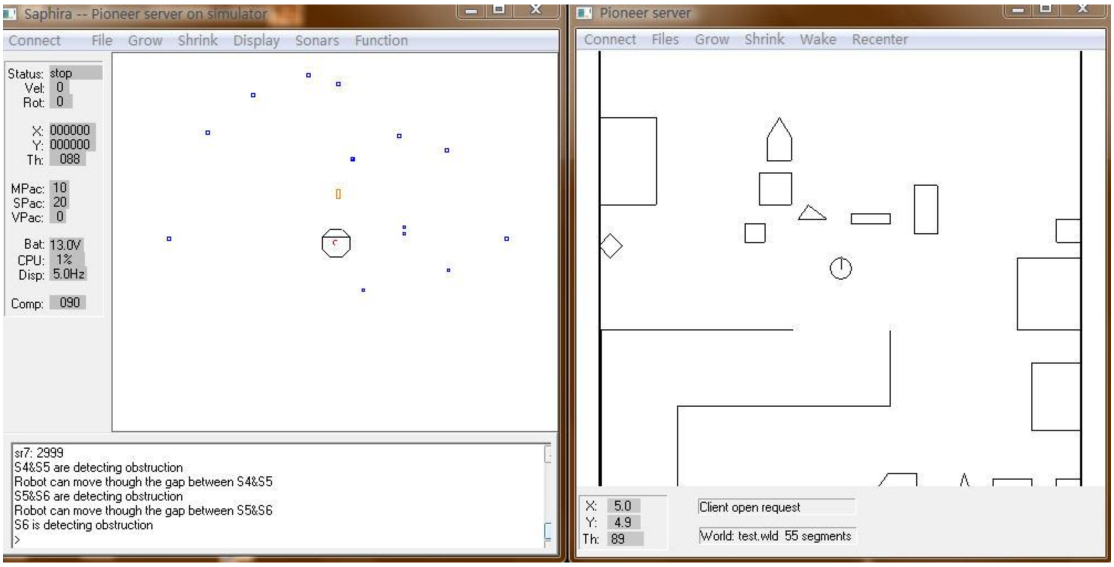


Figure 7.13 Step 5 of Chronis' approach.

In Stage E, see Figure 7.13, the robot meets a cluttered obstruction. The robot is in face of the objects, the instruction of the left window indicates that S4, S5 and S6 are engaged, and the distance between adjacent sensors is shorter than the diameter of the robot.

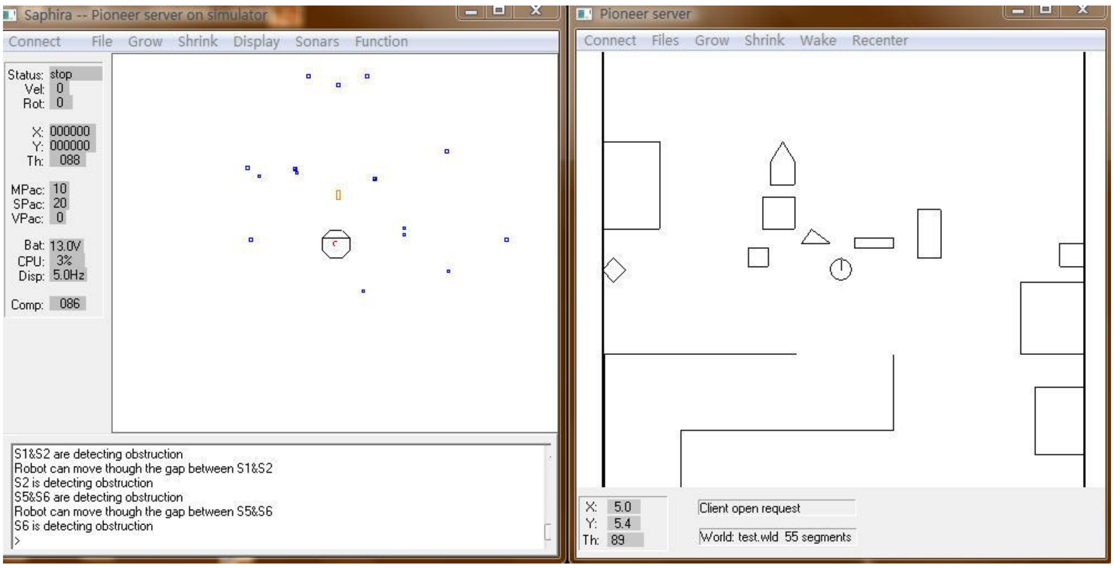


Figure 7.14 Step 6 of Chronis' approach.

As the robot is moving forward, see Figure 7.14, the robot faces the gap between obstructions and the instruction of the left window shows that S3 and S4 are not engaged, because the sensors bounce through the gap. The robot is continuing moving, and this leads the robot to a collision, see Figure 7.15.

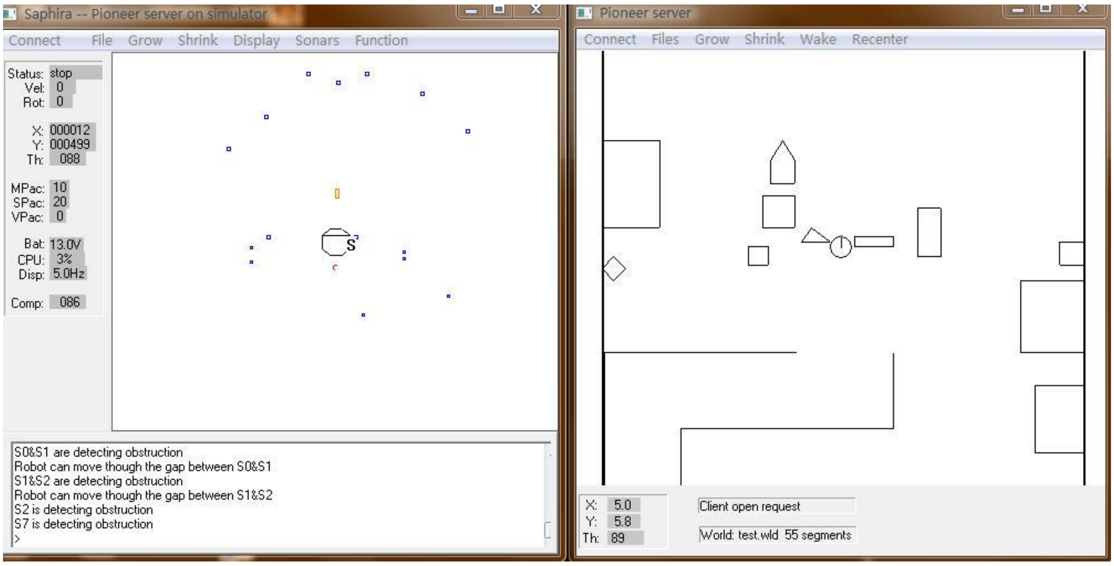


Figure 7.15 Step 7 of Chronis' approach.

Thus, it can be seen that the robot has got stalled in the clutter. The reasons for this are (a) sonar passing through the middle of the objects; and (b) uncertainty in the reading, which suggests no specular reflection in this case. However, if such reflection is present, the robot would be in no better situation. Obviously, this approach is more suitable in a structured environment and it is possible to provide the robot with more information a priori. Sonar specular return problems lead to wrong sonar readings, which subsequently is on its way to result in a wrong detection. Chronis' approach was not a total failure, and there are 4/10 successful runs.

### 7.4.1.2 Statistical Analysis

In this experiment, the total successful goal achievement is 90% of this approach. And in Stage B, there are two sets of trajectory. 40% of trajectories are in the top set and 60 % of trajectories are in the lower set. The deviation of X direction at Y= -2000 is approximately 127 mm. In Stage E, after the robot turns left, the deviation of X direction at Y= -4000 is approximately 249 mm. It was cumulative with the deviations in Stage B. One of nine success stopping points was within 50 mm to the goal points; seven stopping points were within 100 mm, and one stopping point was within 150mm to goal point.

Table 7.15 shows the results of achieving G1. The columns are the same as the previous table. The second column shows the actual decision generated by the robot over the total possible decisions. From the starting point to G1, there are twelve possible decisions generated for the robot: Move Forward, Turn Right, Move Forward, Turn left, Turn Left, Move Forward, Attend Goal, Turn Left, Move Forward, Move Forward, Attend Goal, and Stop.

| | Generated Decisions / Total Decisions | Correct Generated Decisions | Successful runs | Performance | Standard Deviation |
|---|---|---|---|---|---|
| M | 10/12 | 10,9,10,10,9,9,9,10,10,10 | 9/10 | 0.89 | 0.52 |
| C | 10/12 | 4,4,5,4,5,4,9,10,4,4 | 0/10 | 0.69 | 2.26 |

Table 7.15 Results of reaching G1 in Environment 2.

| Structure Detected/Total Possible Structures | Correct Detected Structures | Performance of structure detection | Standard Deviation | Successful runs |
|---|---|---|---|---|
| 8/9 | 8,8,8,8,9,8,8,8,8,8 | 0.88 | 0.32 | 9/10 |

Table 7.16 Results of reaching G1 in Environment 2.

| Structure Detected with Sensor Failure/Total Possible Structures | Correct Detected Structures | Performance of structure detection | Standard Deviation | Successful runs |
|---|---|---|---|---|
| 8/9 | 4,8,8,4,5,5,4,4,8,8 | 0.75 | 1.93 | 4/10 |

Table 7.17 Results of reaching G1 in Environment 2.

### 7.4.2 Test 2: Reaching G2

The Goal 2's coordinates are (8000, -6000) which is located at 2m forward and 7m right of the robot. The robot always starts at (0, 0). During the experiments, to get the trajectory of the robot, the testing program records every coordinate of the robot's movement. The results are shown in the Figure 7.16

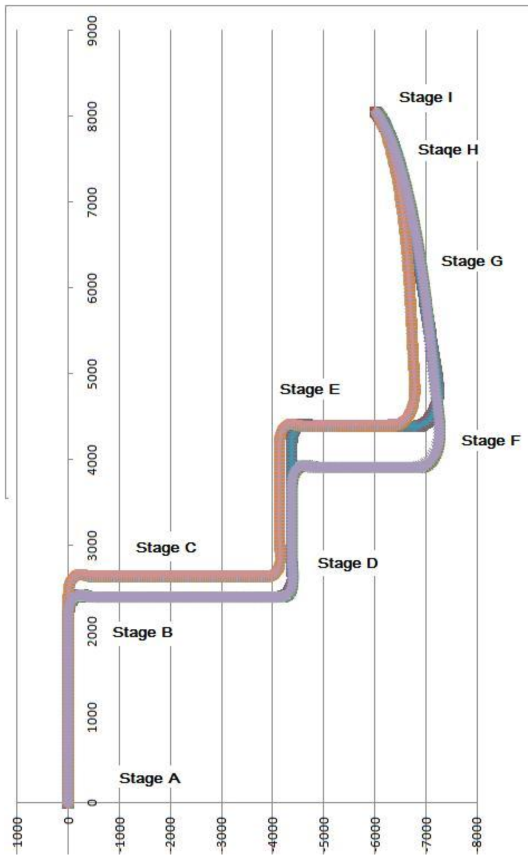


Figure 7.16 Trajectories for reaching G2 in Environment 2

There are nine stages shown in Figure 7.16. In the initial Stage, A, the robot detects the environment as a corridor, suspends the goal attending control, and then keeps moving forward. After the robot exits the corridor, it comes to Stage B. It detects a left corner and decides to turn right. After turning right, in Stage C, the robot detects a corridor. It suspends the goal attending control and moves forward. In Stage D, the robot detects a right corner. The system decides to turn left. After exiting the corner, in Stage E, the robot detects a front wall. As seen from the figure, there are four objects in front of the robot. The robot detects this situation as a possible virtual front wall, which is determined by the detection sensors and the rules. This virtual type of structure is introduced in Chapters 3 and 4. The quadrant system decides to turn right. The routes are driven by two different decisions. The lower routes do not detect any structure and the quadrant system guides the robot to the goal point directly. For the other route in Stage F, the robot detects a front wall. The quadrant system decides to turn right. In Stage G, the robot detects a corridor, and moves forward. In Stage H, after the robot exits the corridor, the quadrant system directs the robot towards the goal point. In Stage I, the robot stops at the goal point.

#### 7.4.2.1 Statistical Analysis

In this experiment the total successful goal achievement is 100%. And in Stage B, there are two sets of trajectory, i.e., 60% of trajectories are in the top set and 40 % of trajectories are in the lower set. The deviation of X direction at Y= -1000 is approximately 124 mm. In Stage E, after the robot turns right, the deviation of X direction at Y= -5000 is approximately 243 mm. Six of ten success stopping points are within 50 mm to goal points; and four stopping points are within 100 mm to goal points.

Table 7.18 shows the results of achieving G2. The columns are the same as the previous table. The second column shows the actual decision generated by the robot over the total possible decisions. From the starting point to G2, there are nine possible decisions generated for the robot: Move Forward, Turn Right, Move Forward, Turn left, Turn Right, Turn right, Move Forward, Attend Goal, and Stop.

| | Generated Decisions / Total Decisions | Correct Generated Decisions | Successful runs | Performance | Standard Deviation |
|---|---|---|---|---|---|
| M | 9/9 | 9,9,9,9,9,9,9,9,9,9 | 10/10 | 1 | 0 |
| C | 5/9 | 4,4,5,4,5,4,5,5,4,4 | 0/10 | 0.87 | 0.56 |

Table 7.18 Results of reaching G2 in Environment 2

| Structure Detected/Total Possible Structures | Correct Detected Structures | Performance of structure detection | Standard Deviation | Successful runs |
|---|---|---|---|---|
| 7/7 | 7,7,7,7,7,7,7,7,7,7 | 1 | 0 | 10/10 |

Table 7.19 Performance of structures detection in reaching G2 in Environment 2

| Structure Detected with Sensor Failure/Total Possible Structures | Correct Detected Structures | Performance of structure detection | Standard Deviation | Successful runs |
|---|---|---|---|---|
| 7/7 | 7,4,4,4,5,7,4,7,7,7 | 0.80 | 1.51 | 5/10 |

Table 7.20 Performance of structure detected with sensors failure in reaching G2 in Environment 2

### 7.4.3 Test 3: Reaching G3

The Goal 3's coordinates is (8000, -8000) which is located at 8m forward and 8m right of the robot. The robot always starts at (0, 0). During the experiments, to achieve the trajectory of the robot, the testing program records every coordinate of the robot's movement, and the results are shown in Figure 7.17.

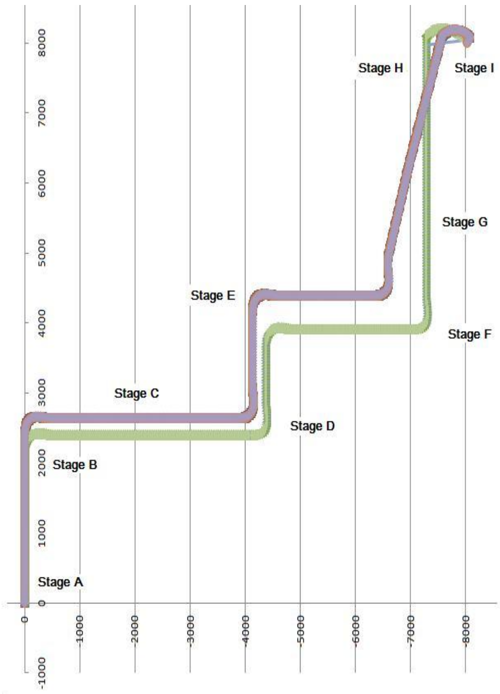


Figure 7.17 Trajectories for reaching G3 in Environment 2

There are nine stages shown in Figure 7.17. In the initial Stage, A, the robot detects the environment as a corridor, suspends the goal attending control, and then keeps moving forward. After the robot exits the corridor, it comes to Stage B. It detects a left corner and decides to turn right. After turning right, in Stage C, the robot detects a corridor, suspends the goal attending control, and moves forward. In Stage D, the robot detects a right corner. The system decides to turn left. After exiting the corner, in Stage E, the robot detects a front wall. As seen from the figure, there are four objects in front of the robot. The robot detects this situation as a possible virtual front wall, which is determined by the detection sensors and the rules. This type of virtual structure is introduced in Chapters 3 and 4. The quadrant system decides to turn right. Also there is a deviation of two sets of routes resulting from the timing for making decisions. In Stage F, the robot detects a front wall. And the quadrant system decides to turn left. In Stage G, there are two sets of routes, in the upper route the robot detects no structure and the quadrant system guides the robot to the goal point. In the lower route, the robot detects a right wall and the quadrant system leads the robot forward. When the robot closes to the goal point, it comes to Stage H. It moves towards to the goal point. In Stage I, the robot stops at the goal point.

#### 7.4.3.1 Statistical Analysis

In this experiment, the total successful goal achievement is 100%. And in Stage B, there were two sets of trajectory, 40% of trajectories in the top set and 60 % of trajectories in the lower set. The deviation of X direction at Y= -1000 is approximately 124 mm. In Stage E, after the robot turns right, the deviation of X direction at Y= -5000 is approximately 248 mm. It was cumulative with the deviations in Stage B. Seven of ten success stopping points were within 50 mm to goal points; and three stopping points were within 100 mm to goal point.

Table 7.21 shows the results of achieving G3. The columns are the same as the previous table. The second column shows the actual decision generated by the robot over the total possible decisions. From the starting point to G3, there are nine possible decisions are generated for the robot: Move Forward, Turn Right, Move Forward, Turn left, Turn Right, Turn Left, Move Forward, Attend Goal, and Stop.

| | Generated Decisions / Total Decisions | Correct Generated Decisions | Successful runs | Performance | Standard Deviation |
|---|---|---|---|---|---|
| M | 9/9 | 9,9,9,9,9,9,9,9,9,9 | 10/10 | 1 | 0 |
| C | 5/9 | 4,4,3,4,4,4,4,5,4,5 | 0/10 | 0.84 | 0.57 |

Table 7.21 Results of reaching G3 in Environment 2

| Structure Detected/Total Possible Structures | Correct Detected Structures | Performance of structure detection | Standard Deviation | Successful runs |
|---|---|---|---|---|
| 7/7 | 7,7,7,7,7,7,7,7,7,7 | 1 | 0 | 10/10 |

Table 7.22 Performance of structures detection in reaching G3 in Environment 2

| Structure Detected with Sensor Failure/Total Possible Structures | Correct Detected Structures | Performance of structure detection | Standard Deviation | Successful runs |
|---|---|---|---|---|
| 7/7 | 7,4,7,4,7,7,4,7,7,4 | 0.81 | 1.55 | 6/10 |

Table 7.23 Performance of structure detected with sensors failure in reaching G3in Environment 2

### 7.4.4 Test 4: Reaching G4

The Goal 4's coordinates is (2000, -7000) which is located at 2m forward and 7m right of the robot. The robot always starts at (0, 0). During the experiments, to achieve the trajectory of the robot, and the testing program records every coordinates of the robot's movement. The results are shown in the Figure 7.18.

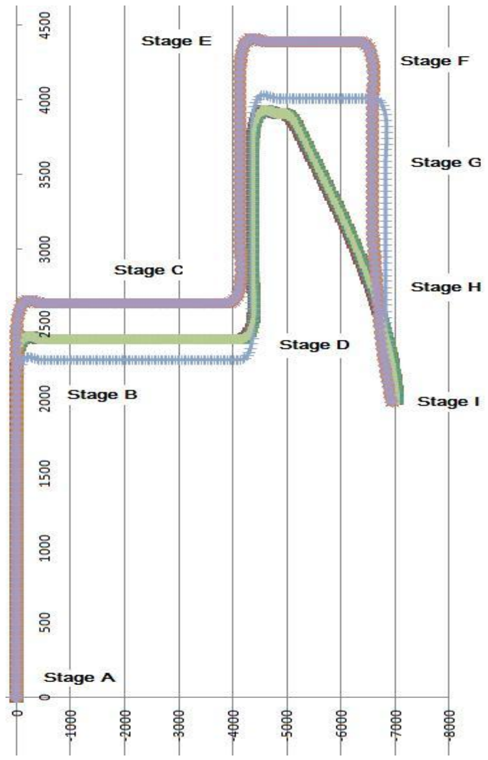


Figure 7.18 Trajectories for reaching G4 in Environment 2

There are nine stages shown in Figure 7.18. In the initial Stage, A, the robot detects the environment as a corridor, suspends the goal attending control, and then keeps moving forward. After the robot exits the corridor, Stage B, it detects a left corner, and decides to turn right. After turning right, in Stage C, the robot detects a corridor, suspends the goal attending control, and moves forward. In Stage D, the robot detects a right corner. The system decides to turn left. After exits the corner, in Stage E, the robot detects a front wall. As seen from Figure 7.7, there are four objects in front of the robot; the robot detects this situation as a possible virtual front wall, which is determined by the detection sensors and the rules. This type of virtual structures is introduced in Chapters 3 and 4. The quadrant system decides to turn right. The routes are driven by two different decisions. The lower routes do not detect any structure. The quadrant system guides the robot to the goal point directly. For the other route in Stage F, the robot detects a front wall. The quadrant system decides to turn right. In Stage G, the robot detects a corridor. It moves forward. In Stage H, after the robot exits the corridor, the quadrant system guides the robot towards the goal point. In Stage I, the robot stops at the goal point.

#### 7.4.4.1 Statistical Analysis

In this experiment, the total successful goal achievement is 100%. And in Stage B, there are two sets of trajectory, 60% of trajectories in the top set and 40 % of trajectories in the lower set. The deviation of X direction at Y= -1000 is approximately 147 mm. In Stage E, after the robot turns right, the deviation of X direction at Y= -5000 is approximately 240 mm. Four of ten success stopping points are within 50 mm to goal points; and six stopping points are within 100 mm to goal point.

Table 7.24 shows the results of achieving G4. The columns are the same as the previous table. The second column shows the actual decision generated by the robot over the total possible decisions. From the starting point to G4, there are nine possible decisions are generated for the robot: Move Forward, Turn Right, Move Forward, Turn left, Turn Right, Turn right, Move Forward, Attend Goal, and Stop.

| | Generated Decisions / Total Decisions | Correct Generated Decisions | Successful runs | Performance | Standard Deviation |
|---|---|---|---|---|---|
| M | 9/9 | 9,9,7,9,7,9,9,9,9,7 | 10/10 | 0.88 | 1.03 |
| C | 6/9 | 6,6,6,6,6,6,6,6,6,6 | 10/10 | 1.0 | 0 |

Table 7.24 Results of reaching G4 in Environment 2

| Structure Detected/Total Possible Structures | Correct Detected Structures | Performance of structure detection | Standard Deviation | Successful runs |
|---|---|---|---|---|
| 7/7 | 7,7,7,7,7,7,7,7,7,7 | 1 | 0 | 10/10 |

Table 7.25 Performance of structures detection in reaching G4 in Environment 2

| Structure Detected with Sensor Failure/Total Possible Structures | Correct Detected Structures | Performance of structure detection | Standard Deviation | Successful runs |
|---|---|---|---|---|
| 7/7 | 7,7,7,7,4,4,7,4,4,7 | 0.84 | 1.55 | 6/10 |

Table 7.26 Performance of structure detected with sensors failure in reaching G4 in Environment 2

### 7.4.5 An Overview of Environment 2 Experimenting Results

Environment 2 is an environment, in which there are cluttered objects in the middle. And it has four goal points, three on the top and the fourth at the bottom. In all the experiments from Stage A to Stage D, all the structures are detected and there is a deviation due to timing of structure detection and the quadrant system's decision making. Of all the experiments, there are 39/40 successful runs by the approach proposed in this thesis. There are 21/40 successful runs with sensors failure. Those failed experiments were mainly in Stage E, when the working sensor bouncing through the gap. By Chronis' approach, there are 10/40 successful runs. The overall performance and success rate are shown in the following table:

| | Average Performance | Success Rate |
|---|---|---|
| M | 0.94 | 0.975 |
| C | 0.85 | 0.25 |

Table 7.27 Overall performance and success rate

Compareing Environment 1 with data in this table, the average performance of Chronis' approach still remain high; however the successful rate drops down. The reason for this in his approach is not suited for a cluttered environment.

The overall structure detection performance and structure detection performance with sensors failure are shown in the following table:

| Average Structure Detection Performance | Average Structure Detection Performance with Sensors Failure |
|---|---|
| 0.97 | 0.8 |

Table 7.28 Performance of average structure detection

### 7.5 Experiments in Environment 3:

Environment 3 is the NEAT research lab of Computer Science, Department of Hull University. The NEAT lab is of 15m by 10.5 m size. There are some tables and chairs in the lab, see Figure 7.19. There are four goal points in this environment.

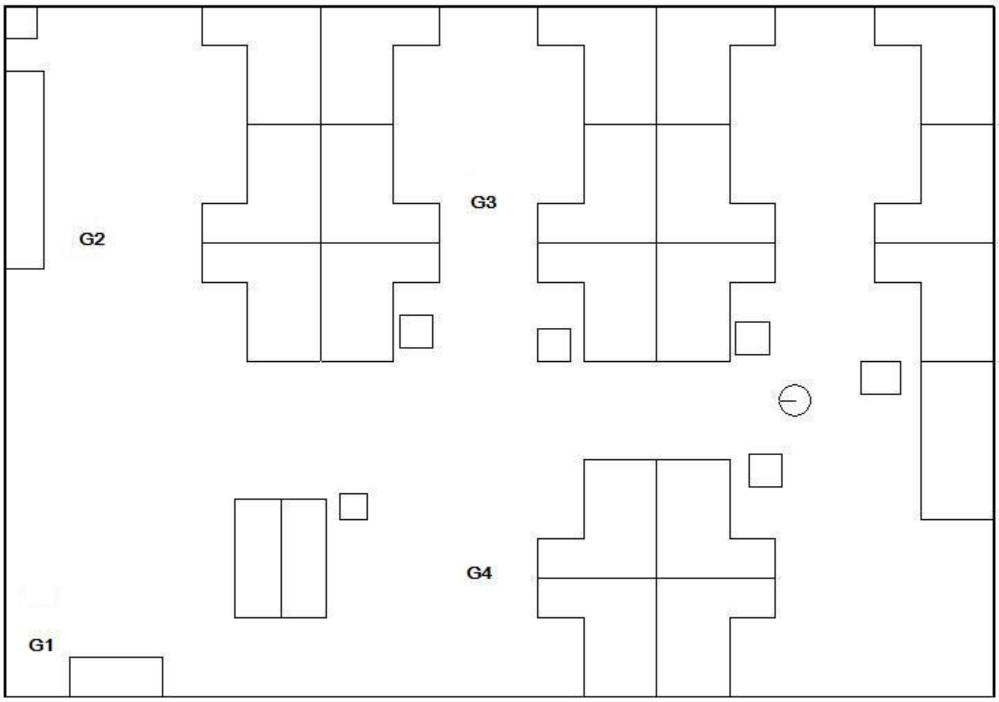


Figure 7.19 Experiment in Environment 3.

#### 7.5.1 Test 1: Reaching G1

The Goal 1's coordinate is (11500, 4000) which is located at 11.5m forward and 4m left of the robot. The robot always starts at (0, 0). During the experiments, to achieve the trajectory of the robot, the testing program records every coordinate of the robot's movement, and the results are shown in Figure 7.20.

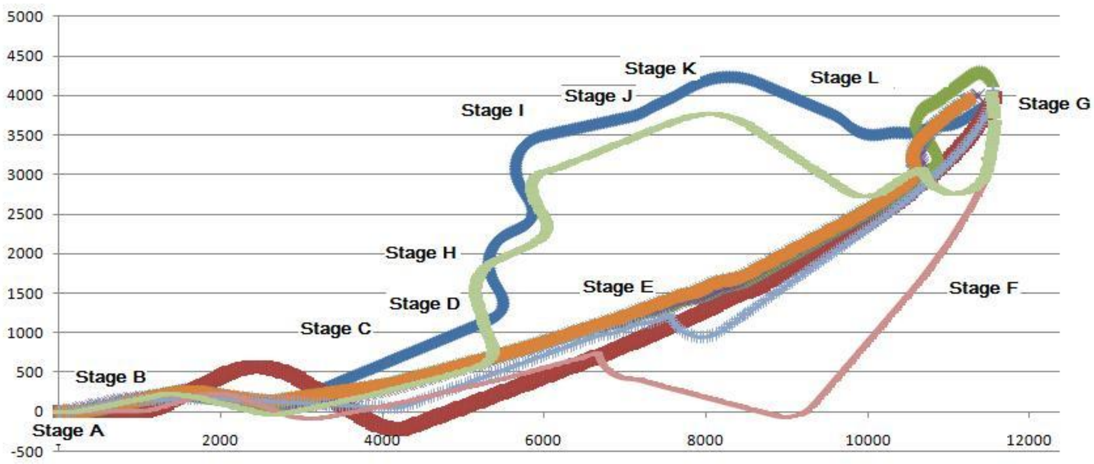


Figure 7.20 Trajectories for reaching G1 in Environment 3

There are twelve stages and two sets of routes in the figure. The routes are driven by different decisions in Stage D. In Stage A, the robot follows the nominal path, which is a straight line to the goal point. When the robot starts to move, the robot enters Stage B immediately. In Stage B, the robot detects a corridor structure, which could be a type of virtual corridor. In Stage B, the quadrant system suspends the goal attending behaviour and the robot moves forward. In Stage C, after the robot exits the corridor, the quadrant system guides the robot towards the goal. There are two sets of routes in Stage D and Stage E respectively. First, In Stage D, the robot detects the obstruction as a front wall, and the quadrant system leads the robot to the left. After the robot turns left, it detects a corridor in Stage H and it moves forward. As seen from the figure, there are some turning and decisions between Stage H and Stage I. When the robot moves from Stage H to I, the corridor structure is no longer detected. And the quadrant system directs the robot to the goal point. The robot detects a front wall, and turns left. In Stage I, the robot detects a front wall, and the quadrant system leads the robot to the right. In Stage J, the robot detects a corridor and moves forward. In Stage K, the robot detects a left corner, and the quadrant system guides the robot to the right. After turning right, in Stage L, the quadrant system directs the robot to the goal point. In Stage G, the robot approaches the goal point, the robot stops at the goal point. By the other route, after Stage C, the robot detects a left wall in Stage E and moves forward. After the robot exits the left wall, the quadrant system leads the robot directly to the goal. Between Stage E and F, there are two routes that make a right turn, see Figure 7.20. Some cases also result in wrong decisions of the robot. The robot closes to the left wall, and S3 had a wrong reading. As a result, the robot detects the condition as a corridor. The quadrant system leads robot to the right. After the robot turns right, the quadrant detects no structure, and then guides the robot to the goal point. Compared with Chronis' work, at the starting point, Stage A, the robot detects three objects located at each side of the robot. The robot cannot direct access goal point; S3 and S4 are in the open range, and the robot is moving forward. In the process, the key decision is made when the robot in open area, which is Stage C. His approach treats the case as one object located in forward left of the robot. And the objects do not cause any obstruction to the goal point. Once the robot is in the open area, the goal attending unit leads the robot towards to the goal point.

#### 7.5.1.1 Statistical Analysis

In this experiment, the total successful goal achievement is 100%. And in Stage D, there are 20% of trajectories, and 80% of trajectories in Stage E. There are 25% of trajectories are made wrong decisions between Stage E and F. The deviation of Y direction at X= 4000 is approximately 217 mm. In Stage E, after the robot turns to right, the deviation of Y direction at X= 6000 is approximately 174 mm. Four of ten success stopping points are within 50 mm to the goal points. Two stopping points are within 100 mm to the goal point, and four stopping points are with 200 mm to the goal points.

Table 7.29 shows the result of achieving G1. The columns are the same as the previous table. The second column shows the actual decision generated by the robot over the total possible decisions. From the starting point to the G1, there are nine possible decisions are generated for the robot: Attend Goal, Move Forward, Turn Left, Turn Right, Move Forward, Turn Right, Attend Goal, Turn Right, Turn right, Move Forward, Attend Goal, and Stop.

| | Generated Decisions / Total Decisions | Correct Generated Decisions | Successful runs | Performance | Standard Deviation |
|---|---|---|---|---|---|
| M | 10/12 | 6,6,10,6,10,6,6,6, 6,6 | 10/10 | 0.77 | 1.69 |
| C | 10/12 | 4,6,4,6,6,4,6,4,6,6 | 6/10 | 0.68 | 1.03 |

Table 7.29 Results of reaching G1 in Environment 3

| Structure Detected/Total Possible Structures | Correct Detected Structures | Performance of structure detection | Standard Deviation | Successful runs |
|---|---|---|---|---|
| 6/8 | 2,2,6,2,6,2,2,2,2,2 | 0.57 | 1.69 | 10/10 |

Table 7.30 Performance of structures detection in reaching G1 in Environment 3

| Structure Detected with Sensor Failure/Total Possible Structures | Correct Detected Structures | Performance of structure detection | Standard Deviation | Successful runs |
|---|---|---|---|---|
| 6/8 | 2,6,2,2,6,4,5,4,6,6 | 0.77 | 1.77 | 7/10 |

Table 7.31 Performance of structure detected with sensors failure in reaching G1 in Environment 3

### 7.5.2 Test 2: Reaching G2

The Goal 2's coordinates is (11000, -2500) which is located at 11m forward and 2.5m right of the robot. The robot always starts at (0, 0). During the experiments, to achieve the trajectory of the robot, and the testing program records every coordinate of the robot's movement. The results are shown in Figure 7.21.

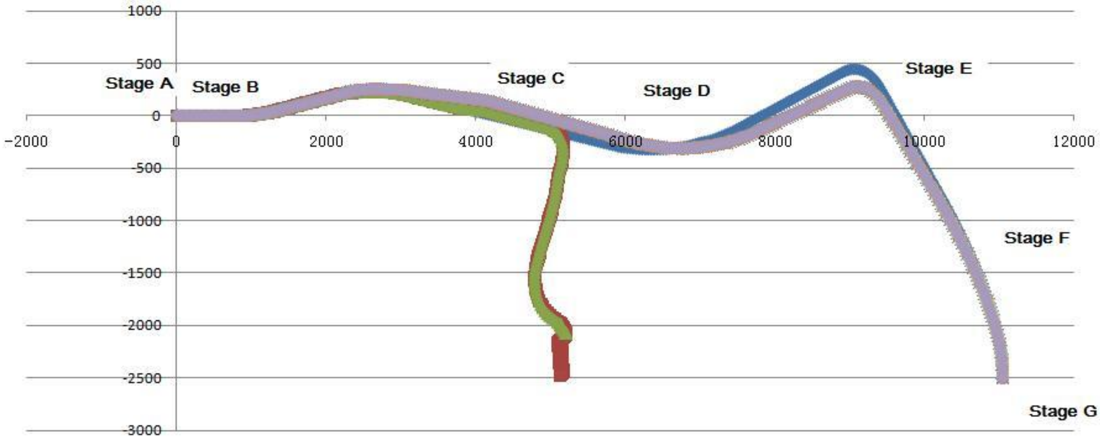


Figure 7.21Trajectories for reaching G2 in Environment 3

There are seven stages in the figure, and there are two sets of routes. The routes are driven by different decisions in Stage C. In Stage A, the robot follows the nominal path, which is a straight line to the goal point. When the robot starts to move, the robot enters Stage B immediately. In Stage B, the robot detects a corridor structure, which could be a type of a virtual corridor. In Stage B, the quadrant system suspends the goal attending behaviour, and the robot moves forward. In Stage C, after the robot exits the corridor, the quadrant system guides the robot towards the goal. In Stage D, the robot detects a corridor and the quadrant system directs the robot forward. After the robot exits the corridor, in Stage E the robot enters an open area. The quadrant system leads the robot directly to the goal. In Stage F, the robot detects a corridor. In travelling in the corridor, the robot approaches the goal point. In Stage G, the robot stops at the goal point. In Stage C, there is a set of routes, turning right early, which lead to the wrong place. It happens while the robot travelling in Stage C, it detects a front wall, and the quadrant system decides to turn right. The decision was made because of S4 receiving a reading. This is a disadvantage of the quadrant system. Compared with Chronis' work, at the starting point, Stage A, the robot detects three objects located at each side of the robot. The robot cannot direct access goal point; S3 and S4 are in the open range, and the robot is moving forward.

In the process, the key decision is made when the robot in open area, which is Stage C. His approach treats the case as one object located in forward left of the robot. And the objects do not cause any obstruction to the goal point. The goal attending unit leads the robot turns right, and robot enters dead-end.

#### 7.5.2.1 Statistical Analysis

In this experiment, the total successful goal achievement is 80%. All of the successes stopping points are within 50 mm to goal points; 20% trajectories fail to reach goal point. Table 7.32 shows the result of achieving G2. The columns are the same as the previous table. The second column shows the actual decision generated by the robot over the total possible decisions. From the starting point to G2, there are seven possible decisions generated for the robot: Attend Goal, Move Forward, Attend Goal, Move Forward, Attend Goal, Move Forward, and Stop.

| | Generated Decisions / Total Decisions | Correct Generated Decisions | Successful runs | Performance | Standard Deviation |
|---|---|---|---|---|---|
| M | 7/7 | 7,7,7,3,7,7,7,3,7,7 | 8/10 | 0.83 | 1.69 |
| C | 3/7 | 3,3,3,3,3,3,3,3,3,3 | 0/10 | 1 | 0 |

Table 7.32 Results of reaching G2 in Environment 3

| Structure Detected/Total Possible Structures | Correct Detected Structures | Performance of structure detection | Standard Deviation | Successful runs |
|---|---|---|---|---|
| 3/3 | 3,3,3,3,3,3,3,3,3,3 | 1 | 0 | 8/10 |

Table 7.33 Performance of structures detection in reaching G2 in Environment 3

| Structure Detected with Sensor Failure/Total Possible Structures | Correct Detected Structures | Performance of structure detection | Standard Deviation | Successful runs |
|---|---|---|---|---|
| 3/3 | 2,3,2,2,3,3,3,3,2,2 | 0.82 | 0.53 | 5/10 |

Table 7.34 Performance of structure detected with sensors failure in reaching G2 in Environment 3

### 7.5.3 Test 3: Reaching G3

The Goal 3's coordinates is (4700, -2800) which is locates at 4.7m forward and 2.8m right of the robot. The robot always starts at (0, 0). During the experiments, to achieve the trajectory of the robot, the testing program records every coordinate of robot movement. The results are shown in Figure 7.22.

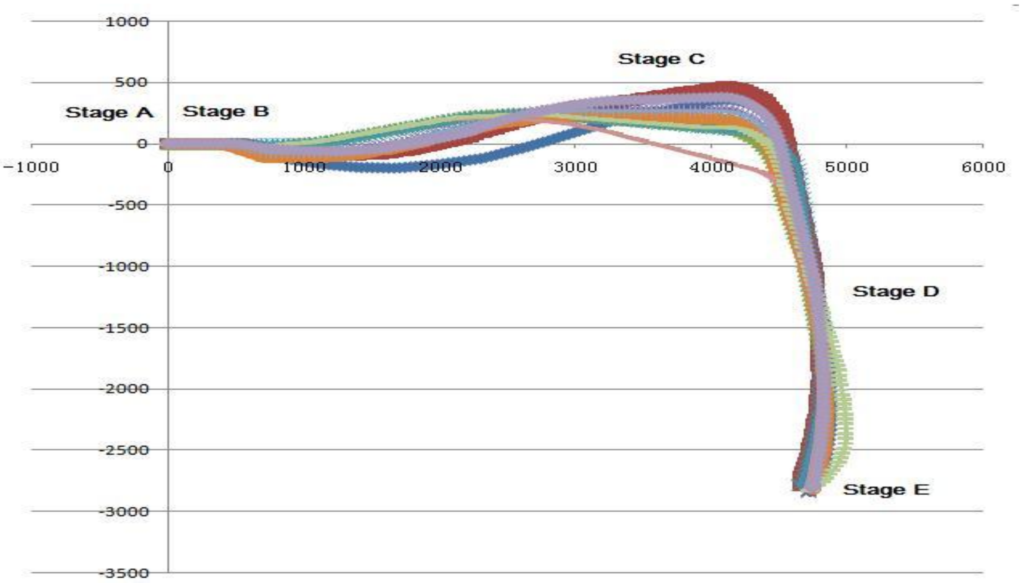


Figure 7.22 Trajectories for reaching G2 in Environment 3

There are five stages in the figure. In Stage A, the robot follows the nominal path, which is a straight line to the goal point. When the robot starts to move, it enters Stage B immediately. In Stage B, the robot detects a corridor structure, which could be a type of a virtual corridor. In Stage B, the quadrant system suspends the goal attending behaviour, and the robot moves forward. In Stage C, after the robot exits the corridor, the quadrant system guides the robot towards the goal. In Stage D, the robot detects a corridor, and the quadrant system directs the robot forward. In Stage E the robot closes a goal point, and stops at the goal point. Compared with Chronis' work, at the starting point, Stage A, the robot detects three objects located at each side of the robot. The robot cannot direct access goal point; S3 and S4 are in the open range, and the robot is moving forward. In the process, the key decision is made when the robot in open area, which is Stage C. His approach treats the case as one object located in forward left of the robot. And the objects do not cause any obstruction to the goal point. The goal attending unit leads the robot turns right and robot achieves the goal point.

#### 7.5.3.1 Statistical Analysis

In this experiment, the total successful goal achievement is 100%. Eight of successes stopping points are within 50 mm to the goal points; two of successes stopping points are with 100mm. Table 7.35 shows the results of achieving G3. The columns are the same as the previous table. The second column shows the actual decision generated by the robot over the total possible decisions. From the starting point to G3, there are five possible decisions generated for the robot: Attend Goal, Move Forward, Attend Goal, Move Forward, and Stop.

| | Generated Decisions / Total Decisions | Correct Generated Decisions | Successful runs | Performance | Standard Deviation |
|---|---|---|---|---|---|
| M | 5/5 | 5,5,5,5,5,5,5,5,5,5 | 10/10 | 1 | 0 |
| C | 5/5 | 5,5,5,5,5,3,5,5,3,3 | 7/10 | 0.84 | 0.97 |

Table 7.35 Results of reaching G2 in Environment 3

| Structure Detected/Total Possible Structures | Correct Detected Structures | Performance of structure detection | Standard Deviation | Successful runs |
|---|---|---|---|---|
| 2/2 | 2,2,2,2,2,2,2,2,2,2 | 1 | 0 | 10/10 |

Table 7.36 Performance of structures detection for reaching G2 in Environment 3

| Structure Detected with Sensor Failure/Total Possible Structures | Correct Detected Structures | Performance of structure detection | Standard Deviation | Successful runs |
|---|---|---|---|---|
| 2/2 | 2,2,2,2,2,2,2,2,2,2 | 1 | 0 | 10/10 |

Table 7.37 Performance of structure detected with sensor failure for reaching G2 in Environment 3

### 7.5.4 Test 4: Reaching G4

The Goal 4's coordinates is (4700, 3300) which is locates at 4.7m forward and 3.3m right of the robot. The robot always starts at (0, 0). During the experiments, to achieve the trajectory of the robot, the testing program records every coordinate of the robot's movement, and the results are shown in Figure 7.23.

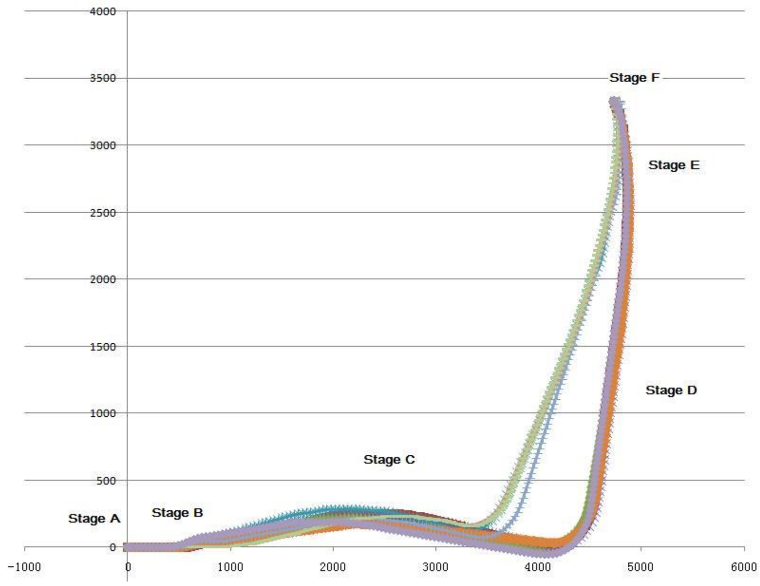


Figure 7.23 Trajectories for reaching G4 in Environment 3

There are six stages in Figure 7.23. In Stage A, the robot follows the nominal path which is a straight line to the goal point. When the robot starts to move, the robot enters Stage B immediately. In Stage B, the robot detects a corridor type structure; this could be a virtual type of corridor. In Stage B, the quadrant system suspends the goal attending behaviour, the robot moves forward. In Stage C, after the robot exits the corridor, the quadrant system guides the robot towards the goal. In Stage D, the robot detects a corridor, and the quadrant system directs the robot forward. In Stage E, the robot detects no structure and the quadrant system guides the robot towards the goal point. In Stage F, the robot closes a goal point, and stops at the goal point. Compared with Chronis' work, at the starting point, Stage A, the robot detects three objects located at each side of the robot. The robot cannot direct access goal point; S3 and S4 are in the open range, and the robot is moving forward. In the process, the key decision is made when the robot in open area, which is Stage C. His approach treats the case as one object located in forward left of the robot. And the objects do not cause any obstruction to the goal point. The goal attending unit leads the robot turns left and robot reaches the goal point.

#### 7.5.4.1 Statistical Analysis

In this experiment, the total successful goal achievement is 100%. Eight of successful stopping points are within 50 mm to goal points; two of successes stopping points are with 100mm. Table 7.38 shows the results of achieving G4. The columns are the same as the previous table. The second column shows the actual decision generated by the robot over the total possible decisions. From the starting point to G4, there are six possible decisions generated for the robot: Attend Goal, Move Forward, Attend Goal, Move Forward, Attend Goal, and Stop.

| | Generated Decisions / Total Decisions | Correct Generated Decisions | Successful runs | Performance | Standard Deviation |
|---|---|---|---|---|---|
| M | 6/6 | 6,6,6,6,6,6,6,6,6,6 | 10/10 | 1 | 0 |
| C | 6/6 | 6,6,6,6,6,6,6,6,6,6 | 10/10 | 1 | 1 |

Table 7.38 Results of reaching G4 in Environment 3

| Structure Detected/Total Possible Structures | Correct Detected Structures | Performance of structure detection | Standard Deviation | Successful runs |
|---|---|---|---|---|
| 2/2 | 2,2,2,2,2,2,2,2,2,2 | 1 | 0 | 10/10 |

Table 7.39 Performance of structures detection for reaching G4 in Environment 3

| Structure Detected with Sensor Failure/Total Possible Structures | Correct Detected Structures | Performance of structure detection | Standard Deviation | Successful runs |
|---|---|---|---|---|
| 2/2 | 2,2,2,2,2,2,2,2,2,2 | 1 | 0 | 10/10 |

Table 7.40 Performance of structure detected with sensor failure for reaching G4 in Environment 3

### 7.5.5 An Overview of Environment 3 Experimenting Results

Environment 3 is a realistic environment, which has tables and chairs segmented in different spaces. There are four goal points, for all experiments attending these four goal points, there are 38/40 successful runs of my approach; there are 22/40 successful runs with sensors failure. And there are 23/40 successful runs of Chronis' approach. The overall performance and success rate are shown in the following table:

| | Average Performance | Success Rate |
|---|---|---|
| M | 0.9 | 0.95 |
| C | 0.81 | 0.575 |

Table 7.41 Overall Performances and Success Rate for Environment 3

The overall structure detection performance and structure detection performance with sensors failure are shown in the following table:

| Average Structure Detection Performance | Average Structure Detection Performance with Sensor Failure |
|---|---|
| 0.89 | 0.90 |

Table 7.42 Performance of Average Structure Detection

### 7.6 Conclusion

There are three different types of environment tested in this project. Environment 1 is a simply structured world, in which both approaches show a better performance and arelatively high number of successful runs. In the second environment, there are cluttered objects in the middle. By Chronis' approach, there is the best performance but low success rates, showing that his approach is not robust in a cluttered situation. On the other hand, the approach proposed in this thesis achieves a high performance and relatively high successful runs. In the last environment, the world is a realistic environment. It is more like a structured cluttered environment. Both approaches perform well. Table 7.43 shows the overall performance and success rates of the new approach proposed here and Chronis' approach. The performance represents the ability and efficiency of the new approach. The success rates represent the effectiveness of the proposed approach here.

| | Overall Performance | Overall Success Rates |
|---|---|---|
| M | 0.92 | 0.87 |
| C | 0.83 | 0.51 |

Table 7.43 Overall Performance and Success Rates

In addition to presenting the robustness of the proposed approach by Table 7.45, structure detection performance with sensors failure and relatively more success rates are shown as well.

| | Structure Detection Performance with Sensor Failure | Success Rates |
|---|---|---|
| M | 0.9 | 0.78 |

Table 7.44 Structure Detection Performance with Sensor Failure and Success Rates.

# Chapter 8 Conclusion and Future work

## 8.1 Conclusion and Contribution

A nominal path is a straight-line path from the starting point to the goal point when in a unknown environment. And the robot relies on fusing sensor information to gain sufficient information to make sense of the current environment. In the navigation process, the robot should be able to detect obstacles and avoid them. The navigation system therefore must be made aware of the changes in the environment by the incoming data gathered from sensors. One of the objectives is to: "Develop an algorithm that can use sonar information and provide more information regarding the environment, in that it should classify environment into structures." The algorithm was designed and developed; the robot uses low level sensors, for example Sonar sensors in this thesis, to classify the environment into structures, see Chapter 3. The algorithm has the innate ability to overcome Sonar uncertainties and sensor failure. Even in the cluttered unknown environment, the approach performs well. The extreme case of the sonar sensors was also established. It was found that up to three sonar sensors at one time can fail, subject to: (a) the crucial sensor 0 and 7 cannot fail at any time; (b) the crucial sensor 3 and 4 cannot fail at same time; (c) two adjacent sonar sensors cannot fail at same time.

In the experiment phase, there are three different types of environment are established: a structured environment, a cluttered environment and a realistic (structured-cluttered) environment. The approach in this thesis performed well in each environment. The efficiency was evaluated using performance measure; the Performance of rules is presented in Chapter 7. The total number of successful runs represents the effectiveness of the approach. In order to show the robustness of the approach, the Performance of structure detection when sensor fails was required. This was answered in Chapter 7 and results were good in terms of efficiency, effectiveness and robustness. In the cluttered environment, the results of the approach in this thesis are significantly better than Chronis work, see Table 7.27; in the structured environment, the results of two approaches are similar, see Table 7.14; and in the last environment, Chronis' approach performs well and the approach in this thesis performs better, see Table 7.41. By comparing these results, it clearly can be seen that the approach in this thesis can survive in both structured and cluttered unknown environments with efficiency, effectiveness and robustness.

The rules enable a robot to navigate in an unknown cluttered environment without help of maps, beacons and high level sensors i.e. vision sensor. This is achieved by using sonar sensors and developed by using little computation and very small amount of memory. Thus the rules demonstrated here would enable a robot with a high payload of power ratio to navigate in difficult environments where little information is available. The contributions of this thesis are:

a) An approach was developed which integrated low level sensory information with robots gaining more efficiency.
b) The extreme conditions for failure of sensors were also established, in the case of the presence of a ring of sonar sensors.
c) The structure detection rules are efficiency while integrated with a quadrant based navigation approach.
d) The approach can be applied to the situation where the environment is not mapped and lacks of proper structure i.e. a cluttered environment.

## 8.2 Future Work

In the experiments, there are certain failed runs caused by the nature of quadrant system. In some situations (such as Environment 3, achieving G3), it leads the robot to the wrong direction and also can cause collision. In the future, researchers should carry on research with several navigation strategies to gain better results. Furthermore, the work in this thesis utilised low level sonars and used the available information efficiently. This was done by improving the process of classification of using rules. However, it is clear that there are important challenges ahead. Low level sonars are important, but they will not provide a complete picture of the environment. This is especially important when 3-D information is required. In such a situation, multiple or omni-directional vision sensors can be fused with these sonar sensors by using the same concept of rules. The problem of fusing the information with these sonar sensors can be considered as a distributed consensus problem. In the future, the concept can be applied in a distributed sensor network area. Indeed, there is work ongoing in the department in this area, and also some work elsewhere, [Bellotto. N, Hu, H, (2009), Calton. J.L, Taube. J.S, (2009), Chong. J.W.S et al (2009), Garcia. M.A.P, et al, (2009), Gu. D, Hu. H (2009), Kim (2009), Nico. D, Daprati. E (2009), Otte et al (2009), Samperio. R,Hu. H,Gu. D, (2009)].

Thus the challenges for the future developments are:

a) Better and more improved optimal navigation strategies have to be developed. Using low level sensory information more efficiency.
b) Integration of higher level sensory information e.g. vision sensor like omni-directional camera, etc.

Finally, the robot navigation is still a changeling area to the researchers, however in the future the researchers will achieve the real intelligence navigation for the robotics.

## References


1. Alonzo Kelly (1996) Introduce to Mobile robot, lecture notes.
2. Arkin.R.C (1990) Integrating behavioral, perceptual and world knowledge in reactive navigation, Robotics and Autonomous Systems, 6, pp.105—122.
3. Arkin,R.C. (1998) Behavior-Based Robotics. MIT Press, Cambridge Massachusetts.
4. Atiya, S. and Hager, G.,(1993) “Real-time Vision-based Robot Localization.” IEEE Transactions on Robotics and Automation, Vol. 9, No. 6, pp. 785-800.
5. Bajcsy. R, Allen. P, (1985) Converging disparate sensory data, The Second International Symposium on Robotics Research, pp. 81-86
6. Barshan, B. and Durrant-Whyte, H.F., (1993) “An Inertial Navigation System for a Mobile Robot.” Proceedings of the 1993 IEEE/RSJ International Conference on Intelligent Robotics and Systems, Yokohama, Japan, July 26-30, pp. 2243-2248.
7. Barshan, B. and Durrant-Whyte, H.F., (1995) “Inertial Navigation Systems Mobile Robots.” IEEE Transactions on Robotics and Automation, Vol. 11, No. 3, June, pp. 328-342.
8. Bellotto. N, Hu, H, (2009), Multisensor-Based Human Detection and Tracking for Mobile Service Robots,IEEE Transactions On Systems, Man, and Cybernetics—Part B: Cybernetics, Vol. 39, No. 1
9. Bertsekas. D, Tsitsiklis. J. N. (1989) Parallel and Distributed Computation: Numerical Methods, Prentice Hall.
10. Bloch. I, (1999) “Fuzzy Relative Position between Objects in Image Processing: New Definition and Properties Based on a Morphological Approach”, Int. J. of Uncertainty Fuzziness and Knowledge-Based Systems, vol. 7, no. 2, pp. 99-133.
11. B. Kuipers, and Y. T. Byun,(1991) "A robot exploration and mapping strategy based on a semantic hierarchy of spatial representations," *Journal of Robotics and Autonomous Systems*,vol. 8, pp. 47-63
12. Borenstein J., Koren Y., (1991) The vector field histogram - fast obstacle avoidance for mobile robots, IEEE Transactions on Robotics and Automation, Vol.7, No.3, pp 278-288

13. Borenstein, J. and Feng. L., (1995) "UMBmark: A Benchmark Test for Measuring Deadreckoning Errors in Mobile Robots." 1995 SPIE Conference on Mobile Robots, Philadelphia, October, pp. 22-26.

14. Borenstein, J. and Feng, L. (1994) "UMBmark & A Method for Measuring, Comparing, and Correcting Dead-reckoning Errors in Mobile Robots." Technical Report, The University of Michigan UM-MEAM-94-22, December.

15. Borenstein, J. and Feng. L. (1996) "Measurement and Correction of Systematic Odometry Errors in Mobile Robots." IEEE Journal of Robotics and Automation, Vol 12, No 5, October.

16. Bruce, J. Veloso, M, (2002) Real-time randomized path planning for robot navigation,Dept. of Comput. Sci., Carnegie Mellon Univ., Pittsburgh, PA, USA,Intelligent Robots and System, Volume: 3, pp. 2383- 2388

17. Byrne, R.H., Klarer, P.R., and Pletta, J.B., (1992) "Techniques for Autonomous Navigation." Sandia Report SAND92-0457, Sandia National Laboratories, Albuquerque, NM, March.

18. Byrne, R.H., (1993) "Global Positioning System Receiver Evaluation Results." Sandia Report SAND93-0827, Sandia National Laboratories, Albuquerque, NM, Sept.

19. Cahn D.F. Phillips S.R., (1975) "ROBNAV: A range-based robot navigation and obstacle avoidance algorithm", IEEE Transactions on Systems, Man and cybernetics, September 1975, pp. 544-551

20. Calton. J.L, Taube. J.S, (2009) Where am I and how will I get there from here? A role for posterior parietal cortex in the integration of spatial information and route planning, Neurobiology of Learning and Memory 91, pp. 186–196

21. C. Freksa, R. Moratz, and T. Barkowsky, (2000) "Schematic Maps for Robot Navigation," in *Spatial Cognition II: Integrating Abstract Theories, Empirical Studies, Formal Methods, and Practical Applications*, C. Freksa, W. Brauer, C. Habel, and K. F. Wender,Eds. Berlin: Springer, pp. 100–114.

22. Chen. Y, Medioni. G, (1991) Object modeling by registration of multiple range images, in: Proceedings of the IEEE International Conference on Robotics and Automation, pp. 2724-2729

23. Chong. J.W.S et al (2009), Robot programming using augmented reality: An interactive method for planning collision-free paths. Robotics and Computer-Integrated Manufacturing 25, pp. 689–701.
24. Chronis.G, (2002) Spatial Relations for Mobile Agent Navigation
25. Chronis.G, (2007) Sketch – Based Navigation For Mobile Robots Using Qualitative Landmark States, A PhD Dissertation presented to the Faculty of the Graduate School University of Missouri-Columbia.
26. Cohen, C. and Koss, F., (1992) "A Comprehensive Study of Three Object Triangulation." Proceedings of the 1993 SPIE Conference on Mobile Robots, Boston, MA, Nov. 18-20, pp. 95-106
27. Connell. J, (1992) SSS: A hybrid architecture applied to robot navigation, in Proceedings of the IEEE Conference on Robotics and Automation, pp. 2719-2724.
28. Cox, I.J., (1991) "Blanche - An Experiment in Guidance and Navigation of an Autonomous Mobile Robot." IEEE Transactions Robotics and Automation, 7(3), pp. 193-204
29. Crowley. J.L. (1985) Navigation of an intelligent mobile robot. IEEE Journal of Robotics Research, 1(1), pp. 31-41.
30. C. Ye, (2009) A sub goal seeking approach for reactive navigation in complex unknown environments, Robotics and Autonomous Systems Volume 57, Issue 9, 30 September 2009, Pages 877-888.
31. D. Dai, and D. T. Lawton, (1993,) "Range-Free Qualitative Navigation,"in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, vol. 1,pp. 783-790
32. D. T. Lawton, R. C. Arkin, and J. M. Cameron, (1990) "Qualitative Spatial Understanding and Reactive Control for Autonomous Robots," in *Proceedings of the International Workshhop on Intelligent Robots and Systems*, Tsuchiura, Japan, vol. 2, pp. 709-714
33. Duan, Q.J. Wang, R.X. Feng, H.S. Wang, L.G, (2004) An immunity algorithm for path planning of the autonomous mobile robot Multitopic Conference, Proceedings of INMIC 2004. 8th International, Publication Date: 24-26 Dec. 2004,On page(s): 346- 350.
34. Dudek.G, Jenkin.M, (2000) Computational Principles of Mobile Robotics,

Cambridge University Press, United Kingdom

35. Freeman. J. (1975) , “The Modelling of Spatial Relations”, Computer Graphics and Image Processing (4), pp. 156-171.
36. Garcia. M.A.P, et al, (2009) Path planning for autonomous mobile robot navigation with ant colony optimization and fuzzy cost function evaluation, Applied Soft Computing 9, pp. 1102–1110.
37. Gat. E, (1992) Integrating planning and reacting in a heterogeneous asynchronous architecture for controlling real-world mobile robots, in Proceedings of the AAAI Conference.
38. Getting, I.A., (1993) “The Global Positioning System,” IEE Spectrum, December, pp. 36-47.
39. Ghatee. M, Mohades. A, (2009) Motion planning in order to optimize the length and clearance applying a Hopfield neural network, Expert Systems with Applications Volume 36, Issue 3, Part 1, pp. 4688-4695
40. GPS Report. November 5, 1992. Potomac, MD: Phillips Business Information.
41. Gribble. W, Browning. R, Hewett. M, Remolina. E and Kuipers. B, (1998) “Integrating vision and spatial reasoning for assistive navigation”, In Assistive Technology and Artificial Intelligence, V. Mittal, H. Yanco, J. Aronis and R. Simpson, ed., Springer Verlag, pp. 179-193.
42. Gu. D, Hu. H (2009), Distributed network-based formation control, International Journal of Systems Science Vol. 40, No.5, pp. 539–552.
43. Humphrys, M. (1997) Action Selection Methods Using Reinforcement Learning, Ph.D. dissertation, University of Cambridge.
44. I. R. Nourbakhsh, A. Soto, J. Bobenage, S. Grange, R. Meyer, and R. Lutz, (1999) "An effective mobile robot educator with a full-time job," *Artificial Intelligence*, vol. 114, no. 1-2, pp. 95-124
45. Iyengar S.S., Jorgensen C.C., Rao S.V.N., Weisbin C.R., (1987) “Robot Navigation in Unknown Terrains Using Learned Visibility Graphs. Part I: The Disjoint Convex Obstacle Case”, IEEE Transactions on Robotics and Automation, Vol RA-3, No.6 December 1987, pp 672-681.

46. Jadbabaie. A, Lin. J, and Morse. A. S, (2003) Coordination of groups of mobile autonomous agents using nearest neighbor rules,. IEEE Transactions on Automatic Control, Vol. 48, No. 3, pp. 988-1001.
47. Jaafar. J. McKenzie. E, (2008) A Fuzzy Action Selection Method for Virtual Agent Navigation in Unknown Virtual Environments, Journal of Uncertain Systems Vol.2, No.2, pp.144-154.
48. J. Cao (1999) Reactive navigation for autonomous guided vehicle using neuron-fuzzy techniques, pp.108-117
49. J.Ng, (2010) An analysis of mobile robot navigation algorithms in unknown environments. PhD thesis, The University of Western Australia School of Electrical, Electronic and Computer Engineering, M018 35 Stirling Highway Crawley, Perth, WA 6009
50. Kak, A., Andress, K., Lopez-Abadia, and Carroll, M., (1990) "Hierarchical Evidence Accumulation in the PSEIKI System and Experiments in Model-driven Mobile Robot Navigation." Uncertainty in Artificial Intelligence, Vol. 5, Elsevier Science Publishers B. V., North- Holland, pp. 353-369.
51. Kambhampati C, Xiao. Y. (2003) Localisation and planning for Landmine detection Proc. IEEE SMC
52. Kayton.M (1989) Navigation: Land, Sea, Air, Space. IEEE Press,
53. Kim. D. H, (2009) Escaping Route Method for a Trap Situation in Local Path Planning International Journal of Control, Automation, and Systems (3), pp. 495-500
54. Kleeman. L, Kuc. R, (1995) Mobile robot sonar for target localization and classification, Int. J. Robotics Res. 14, pp. 295–318.
55. Konolige. K, Myers. K, Saffiotti. A and Ruspini. E, (1997) The Saphira architecture: a design for autonomy, Journal of Experimental and Theoretical Artificial Intelligence, pp. 215--235.
56. Kuc. R and Siegel .M .W (1987) Physically based simulation model for acoustic sensor robot navigation. IEEE Transactions on Pattern Analysis and Machine Intelligence, pp. 766-778.
57. Latombe, J.C. (1991), Robot Motion Planning, Kluwer Academic Publishers

58. Latombe, J.C. (1993). *Robot motion planning*. New York: Kluwer Academic.
59. Leonard. (1992) Directed Sonar Sensing For Mobile Robot Navigation, Kluwer Academic Publishers, USA
60. Linder, S. P. (1998). Behavior Based Robust Fuzzy Control. In Proceedings of the 1998 IEEE ISIC/CIRA/ISAS Joint Conference, Gaithersburg, Maryland, USA, pp. 233.
61. Liu. Y, Yang. Y.R. (2003) .Reputation Propagation and Agreement in Wireless Ad Hoc Networks,. Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC).
62. Lumelsky V.L, Stepanov A.A, (1987) “Path- Planning strategies for a point mobile automaton moving amidst unknown obstacles of arbitrary shape ”, algorthmica, 1987, 2, pp. 403-430
63. M.A. Abidi, R.C. Gonzalez, (1992) Data Fusion in Robotics and Machine Intelligence, Academic Press, New York,
64. Mehyar. M, Spanos. D, Pongsajapan. J, Low. S.H, Murray. R.M, (2005) Distributed Averaging on Asynchronous Communication Networks, Proceedings of the Joint 44th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC'05), Seville, Spain.
65. Miyajima,K, Ralescu. A, (1994) “Spatial Organization in 2D Segmented Images: Representation and Recognition of Primitive Spatial Relations”, Fuzzy Sets and Systems, vol. 65, iss. 2/3, pp. 225-236.
66. Morgan A. V. and Russel P. I. (1998) What on Earth Achieves.
67. Nico. D, Daprati. E (2009), The egocentric reference for visual exploration and orientation, Brain and Cognition 69, pp. 227- 235
68. Otte. M. W, Richardson. S.G, Mulligan.J, and Grudic. G, (2009) Path Planning in Image Space for Autonomous Robot Navigation in Unstructured Environments Journal of Field Robotics 26(2), pp. 212–240
69. Oskoei. M. A, Gan. J.Q, and Hu.H, (2009) Adaptive schemes applied to online SVM for BCI data classification, The 31st Annual Int. Conf. of the IEEE Engineering in Medicine & Biology Society, Hilton, Minneapolis, Minnesota, 2-6 September 2009.

70. Oskoei. M.A. and Hu.H, (2009) Adaptive myoelectric Human-Machine Interface for video games, Proc. of IEEE International Conference on Mechatronics and Automation, Changchun, pp. 9-12

71. Payton. D. W. Rosenblatt. J. K., and Keirsey D.M. 1990 Plan guided reaction, IEEE Trans. on Systems, Man, and Cybernetics 20 (6).

72. P. E. Michon, and M. Denis, (2001) “When and Why Are Visual Landmarks Used in Giving Directions?” in Spatial Information Theory: Foundations of Geographic Information Science, D. R. Montello, Ed. Berlin: Springer, pp. 292-305.

73. Perzanowski. D, Schultz. A, Adams. W and Marsh. E, (1999) “Goal Tracking in a Natural Language Interface: Towards Achieving Adjustable Autonomy”, In Proceedings of the 1999 IEEE International Symposium on Computational Intelligence in Robotics and Automation, Monterey, CA, pp. 208-213.

74. Porrill. J, (1988) Optimal combination and constraints for geometrical sensor data, The International Journal of Robotics Research 7 (6), pp. 66-77

75. Rau.A.V, Dunningham. J. A, Burnett. K. (2003) Measurement-Induced Relative-Position Localization Through Entanglement, Science 22:Vol. 301. no. 5636, pp. 1081 – 1084

76. R. Muller, T. Rofer, A. Lankenau, A. Musto, K. Stein, and A.Eisenkolb (2000) “Coarse Qualitative Descriptions in Robot Navigation,” in *Spatial Cognition II: Lecture Notes in Artificial Intelligence*, C. Freksa, W. Brauer, C. Habel, and K. F. Wender,Eds. Berlin: Springer, pp. 265-276.

77. Saffotti, A. (1998) Fuzzy logic in Autonomous Robot Navigation: A case study, ser. Handbook of Fuzzy Computation, Oxford Univ. Press and IOP Press, Oxford, UK.

78. Saffiotti. A, (2000) Fuzzy logic in autonomous navigation, in: D. Driankov, A. Saffiotti (Eds.), Fuzzy Logic Techniques for Autonomous Vehicle Navigation, Physica-Verlag, Heidelberg, New York, pp. 3–22.

79. Samperio. R,Hu. H,Gu. D, (2009), Implementation of A Localization-oriented HRI For Walking Robots In The Robocup Environment, International Journal of Information Acquisition Vol. 5, No. 4, pp. 1–17.

80. Seraji. H, Howard. (2002) Behavior-based robot navigation on challenging terrain: A fuzzy logic approach, IEEE Transactions on Robotics and Automation 18 (3), pp.

308–321.

81. Sgorbissa, A, Arkin. R. C, (2003) Local navigation strategies for a team of robots. Robotica, pp. 461-473.
82. Shibata. F, Ashida. M, Kakusho. K, Babaguchi. N, and Kitahashi. T, (1996) “Mobile Robot Navigation by User-Friendly Goal Specification”, In Proceedings of the 5th IEEE International Workshop on Robot and Human Communication, Tsukuba, Japan, pp. 439-444.
83. Spiers. A J, Warwick. K, Gasson. M N, Ruiz. V F (2007), "Issues impairing the Success of neural implant technology", Applied Bionics and Biomechanics, Vol.3, Issue.4, pp.297-304.
84. Stachniss. C, Burgard. W, (2003) ,Exploring Unknown Environments with Mobile Robots using Coverage Maps, University of Freiburg, Department of Computer Science, D-79110 Freiburg, Germany.
85. Stopp. E, Gapp. K.P, Herzog. G, Laengle. T and Lueth. T, (1994) “Utilizing Spatial Relations for Natural Language Access to an Autonomous Mobile Robot”, In Proceedings of the 18th German Annual Conference on Artificial Intelligence, Berlin, Germany, pp. 39-50.
86. S. Simhon, and G. Dudek, “A Global Topological Map formed by Local Metric Maps,” in *Proceedings of the IEEE International Conference on Intelligent Robots and Systems (IROS),* Victoria, BC, Canada, 13-17 October 1998, vol. 3, pp. 1708-1714.
87. Talluri, R., and Aggarwal, J., (1993) “Position Estimation Techniques for an Autonomous Mobile Robot - a Review.” in Handbook of Pattern Recognition and Computer Vision, World Scientific: Singapore, Chapter 4.4, pp. 769-801
88. T. Lozano-Perez, M.A. Wesley, (1979) An algorithm for planning collision-free paths among polyhedral obstacles, Commun. ACM 22 (10), pp. 560-570.
89. Tan.G, He H, Aaron. S, (2006) Global optimal path planning for mobile robot based on improved Dijkstra algorithm and ant system algorithm, Journal Of Central South University Of Technology,Vol.13 No.1 P.80-86
90. Tompkins. P, (2005) MISSION-DIRECTED PATH PLANNING FOR PLANETARY ROVER EXPLORATION, The Robotics Institute Carnegie Mellon University

Pittsburgh, PA 15213

91. Tsitsiklis. J. N. (1984) Problems in Decentralized Decision Making and Computation,. Ph.D. Thesis, Department of EECS, MIT.
92. Wang. M, Liu J.N.K. (2004) Online path searching for robot autonomous navigation, in: IEEE Conference on Robotics, Automation and Mechatronics, RAM, Singapore, pp. 746–751
93. Wang. M, Liu. J.N.K, (2008) Fuzzy logic-based real-time robot navigation in unknown environment with dead ends, Robotics and Autonomous Systems 56, pp. 625–643
94. Warwick. K, Nasuto. S J (2007), "Rational AI: What does it mean for a machine to be intelligent?", IEEE Instrumentation and Measurement Magazine, Vol.9, Issue.6, pp.20-26.
95. Warwick.K (2007), "The Promise and Threat of Modern Cybernetics", Southern Medical Journal, Vol.100, Issue.1, pp.112-115.
96. Washington. R, Golden. K, Bresina. J, Smith. D.E, Anderson. C, Smith. T, (1999) Autonomous Rovers for Mars Exploration NASA Ames Research Center, MS 269-2 Moffett Field, CA 94035 650-604-5000.
97. Weng.L, Song,D. Y. (2005) Path planning and path tracking control of unmanned ground vehicles (UGVs) Dept. of Electr. Eng., North Carolina A&T State Univ., USA; System Theory. SSST '05. Proceedings of the Thirty-Seventh Southeastern Symposium on, pp. 262- 266
98. Wijk.O, Christensen.H.I., (2000) Localization and navigation of a mobile robot using natural point landmarks extracted from sonar data, Robotics Autonomous Systems 31, pp. 31–42.
99. Ye. C, Wang. D, (2000) A novel behavior fusion method for the navigation of mobile robots, in: IEEE International Conference on SMC, vol. 5, pp. 3526–3531
100. Yen J., Pfluger N., (1995) A Fuzzy Logic Based Extension to Payton and Rosenblattt’s Command Fusion Method for Mobile Robots, IEEE Transactions on Systems, Man, and Cybernetics, Vol 25, No 6, June 1995, pp. 971-978

## Bibliography


1. Abidi, M.A, Gonzalez. R.C., (1992) Data Fusion in Robotics and Machine Intelligence, Academic Press, New York,
2. Adams, M. et al., (1994), "Control and Localization of a Post Distributing Mobile Robot." 1994 International Conference on Intelligent Robots and Systems (Laos '94). München, Germany, September 12-16, pp. 150-156.
3. Alonzo.K, (1996), Introduce to Mobile robot, lecture notes.
4. Anderson.T, and Lee. P.A., (1981), Fault Tolerance Principles and Practice. Englewood Cliffs, NJ: Prentice Hall.
5. Arkin.R.C (1990) Integrating behavioral, perceptual and world knowledge in reactive navigation, Robotics and Autonomous Systems, 6, pp.105—122.
6. Arkin, R. C. (1998). Behavior-Based Robotics. MIT Press, Cambridge Massachusetts.
7. Armingol. J.M., Moreno. L.E., Escalera. A.de la and Salichs. M.A., (2002),A genetic algorithm for mobile robot localization using ultrasonic sensors, Journal of Intelligent and Robotic Systems
8. Atiya, S. and Hager, G., (1993), “Real-time Vision-based Robot Localization.” IEEE Transactions on Robotics and Automation, Vol. 9, No. 6, pp. 785-800.
9. Bajcsy. R, Allen. P, (1985) Converging disparate sensory data, The Second International Symposium on Robotics Research, pp. 81-86
10. Ballard.D and Brown.C (1982) Computer Vision. Prentice-Hall.
11. Barshan, B. and Durrant-Whyte, H.F., (1993), “An Inertial Navigation System for a Mobile Robot.” Proceedings of the 1993 IEEE/RSJ International Conference on Intelligent Robotics and Systems, Yokohama, Japan, July 26-30, pp. 2243-2248.
12. Barshan, B. and Durrant-Whyte, H.F., (1995), “Inertial Navigation Systems Mobile Robots.” IEEE Transactions on Robotics and Automation, Vol. 11, No. 3, June, pp. 328-342.
13. Baskent.D and Barshan.B, (1999) Surface profile determination from multiple sonar data using morphological processing. The International Journal of Robotics Research,

18(8), pp. 788-808.

14. Bellotto. N, Hu, H, (2009), Multisensor-Based Human Detection and Tracking for Mobile Service Robots,IEEE, Transactions on Systems, Man and Cybernetics-part B: Cybernetics, vol, 39, No.1
15. Borenstein J., Koren Y., (1991), The vector field histogram - fast obstacle avoidance for mobile robots, IEEE Transactions on Robotics and Automation, Vol.7, No.3, pages 278-288, 1991
16. Bertsekas. D, Tsitsiklis. J. N. (1989) Parallel and Distributed Computation: Numerical Methods, Prentice Hall.
17. Bloch. I, (1999) “Fuzzy Relative Position between Objects in Image Processing: New Definition and Properties Based on a Morphological Approach”, Int. J. of Uncertainty Fuzziness and Knowledge-Based Systems, vol. 7, no. 2, pp. 99-133.
18. Borenstein, J. and Feng, L., (1994), “UMBmark & A Method for Measuring, Comparing, and Correcting Dead-reckoning Errors in Mobile Robots.” Technical Report, The University of Michigan UM-MEAM-94-22, December.
19. Borenstein, J. and Feng. L., (1995), “UMBmark: A Benchmark Test for Measuring Deadreckoning Errors in Mobile Robots.” 1995 SPIE Conference on Mobile Robots, Philadelphia, pp. 22-26.
20. Borenstein, J. and Feng. L., (1996), “Measurement and Correction of Systematic Odometry Errors in Mobile Robots.” IEEE Journal of Robotics and Automation, Vol 12, No 5, October.
21. Borenstein. J, Everett. H.R., Feng.L, and Wehe.D, (1996) Mobile Robot Positioning-Sensors and Techniques, Journal of Robotic Systems, Special Issue on Mobile robots. Vol, 14 No. 4, pp.231-249,
22. Bruce, J. Veloso, M, (2002) Real-time randomized path planning for robot navigation,Dept. of Comput. Sci., Carnegie Mellon Univ., Pittsburgh, PA, USA,Intelligent Robots and System, 2002,Volume: 3, pp. 2383- 2388
23. Byrne, R.H., Klarer, P.R., and Pletta, J.B., (1992), “Techniques for Autonomous Navigation.” Sandia Report SAND92-0457, Sandia National Laboratories, Albuquerque, NM, March.

24. Byrne, R.H., (1993), “Global Positioning System Receiver Evaluation Results.” Sandia Report SAND93-0827, Sandia National Laboratories, Albuquerque, NM, Sept.

25. Calton. J.L, Taube. J.S, (2009) Where am I and how will I get there from here? A role for posterior parietal cortex in the integration of spatial information and route planning, Neurobiology of Learning and Memory 91,pp. 186–196

26. Cahn D.F. Phillips S.R.,( 1975), “ROBNAV: A range-based robot navigation and obstacle avoidance algorithm”, IEEE Transactions on Systems, Man and cybernetics, September 1975, pp. 544-551

27. Cao. J, Liao. X, and Hall.E.L, Reactive navigation for autonomous guided vehicle using neuron-fuzzy techniques, pp.108-117

28. Chen. Y, Medioni. G, (1991) Object modeling by registration of multiple range images, in: Proceedings of the IEEE International Conference on Robotics and Automation, pp. 2724-2729

29. Chong. J.W.S et al (2009), Robot programming using augmented reality: An interactive method for planning collision-free paths. Robotics and Computer-Integrated Manufacturing 25, pp. 689–701.

30. Chronis.G, (2002) Spatial Relations for Mobile Agent Navigation

31. Chronis.G, (2007) Sketch – Based Navigation For Mobile Robots Using Qualitative Landmark States, A PhD Dissertation presented to the Faculty of the Graduate School University of Missouri-Columbia, 2007

32. Cohen, C. and Koss, F., (1992), “A Comprehensive Study of Three Object Triangulation.” Proceedings of the 1993 SPIE Conference on Mobile Robots, Boston, MA, Nov. 18-20, pp. 95-106.

33. Connell. J, (1992) SSS: A hybrid architecture applied to robot navigation, in Proceedings of the IEEE Conference on Robotics and Automation, pp. 2719-2724.

34. Cox, I.J., (1991), “Blanche - An Experiment in Guidance and Navigation of an Autonomous Mobile Robot.” IEEE Transactions Robotics and Automation, 7(3), pp. 193-204.

35. Crowley J.L., (1985), “Navigation for an intelligent robot”, IEEE Transactions on

Robotics and Automation, Vol. 1, March 1985, pp. 31-41.

36. Crowley, J.L. and Reignier, P., (1992), "Asynchronous Control of Rotation and Translation for a Robot Vehicle." Robotics and Autonomous Systems, Vol. 10, 1992, pp. 243-251.
37. De Silva, C.W. and MacFarlane, A.G.J. (1989a). Knowledge-based control with Application to Robots, Springer-verlag, Berlin.
38. De Silva, C.W. and MacFarlane, A.G.J. (1989b). Knowledge-based control approach for robotic manipulators, Int. J. Control, Vol. 50, No.1, pp 249-273
39. Demirli.K, TMurkNsen.I, (1996) Fuzzy logic based mobile robot localization with sonar data, in: Canada–Japan Bilateral Workshop on Intelligent Manuf. and Process, Design, Toronto, Canada, pp. 28–30.
40. Demirli.K, TMurkNsen.I, (2000) Sonar based mobile robot localization by using fuzzy triangulation, Robotics Autonomous Systems 33 (2–3), pp.109–123.
41. Demirli.K, Molhim. M, (2004), Fuzzy dynamic localization for mobile robots, Fuzzy Sets and Systems 144 , pp. 251–283
42. Duan, Q.J. Wang, R.X. Feng, H.S. Wang, L.G, (2004) An immunity algorithm for path planning of the autonomous mobile robot Multitopic Conference, Proceedings of INMIC 2004. 8th International,Publication Date: 24-26 Dec. 2004, pp. 346- 350
43. Dudek.G, Jenkin.M, (2000) Computational Principles of Mobile Robotics, Cambridge University Press, United Kingdom,
44. Freeman. J. (1975) , “The Modelling of Spatial Relations”, Computer Graphics and Image Processing (4), pp. 156-171.
45. Garcia. M.A.P, et al, (2009) Path planning for autonomous mobile robot navigation with ant colony optimization and fuzzy cost function evaluation, Applied Soft Computing 9, pp. 1102–1110.
46. Gat. E, (1992) Integrating planning and reacting in a heterogeneous asynchronous architecture for controlling real-world mobile robots, in Proceedings of the AAAI Conference.
47. GasKos. J,MartKLn. A, (1996) Mobile robot localization using fuzzy maps, in: A.

Ralescu, T. Martin (Eds.), Lecture Notes in Arti.cial Intelligence, Springer, Berlin.

48. Getting, I.A., (1993), “The Global Positioning System,” IEE Spectrum, December, pp. 36-47.
49. Ghatee. M, Mohades. A, (2009) Motion planning in order to optimize the length and clearance applying a Hopfield neural network, Expert Systems with Applications Volume 36, Issue 3, Part 1, April 2009, Pages 4688-4695. Department of Computer Science, Amirkabir University of Technology, No. 424, Hafez Ave., Tehran 15875-4413, Iran
50. Gourley, C. and Trivedi, M., (1994), "Sensor Based Obstacle Avoidance and Mapping for Fast mobile Robots." Proceedings of the 1994 IEEE International Robotics and Automation, San Diego, CA, May 8-13, pp. 1306-1311.
51. Gribble. W, Browning. R, Hewett. M, Remolina. E and Kuipers. B, (1998), “Integrating vision and spatial reasoning for assistive navigation”, In Assistive Technology and Artificial Intelligence, V. Mittal, H. Yanco, J. Aronis and R. Simpson, ed., Springer Verlag, , pp. 179-193
52. Gu. D, Hu. H (2009), Distributed network-based formation control, International Journal of Systems Science Vol. 40, No. 5, pp. 539–552.
53. Hao Ying, 2000. Fuzzy Control and Modeling: Analytical Foundations and Applications, Wiley-IEEE Press
101. Humphrys, M. (1997) Action Selection Methods Using Reinforcement Learning, Ph.D. dissertation, University of Cam-bridge.
54. Iyengar S.S., Jorgensen C.C., Rao S.V.N., Weisbin C.R., (1987), “Robot Navigation in Unknown Terrains Using Learned Visibility Graphs. Part I: The Disjoint Convex Obstacle Case”, IEEE Transactions on Robotics and Automation, Vol RA-3, No.6 December 1987, pp. 672-681.
102. Jadbabaie. A, Lin. J, and Morse. A. S, (2003) Coordination of groups of mobile autonomous agents using nearest neighbor rules,. IEEE Transactions on Automatic Control, Vol. 48, No. 3, pp. 988-1001.
55. Jaafar. J. McKenzie. E, (2008) A Fuzzy Action Selection Method for Virtual Agent Navigation in Unknown Virtual Environments, Journal of Uncertain Systems Vol.2,

No.2, pp.144-154

56. Jefferies. M.E. and Yeap.W.K., Smith.L, Ferguson. D, (2002), BUILDING A MAP FOR ROBOT NAVIGATION USING A THEORY OF COGNITIVE MAPS.
57. Kak, A., Andress, K., Lopez-Abadia, and Carroll, M., (1990), “Hierarchical Evidence Accumulation in the PSEIKI System and Experiments in Model-driven Mobile Robot Navigation.” Uncertainty in Artificial Intelligence, Vol. 5, Elsevier Science Publishers B. V., North- Holland, pp. 353-369.
58. Kambhampati C, Xiao. Y. (2003) Localisation and planning for Landmine detection Proc. IEEE SMC
59. Kant K. Zucker S., (1988), Planning collision free trajectories in time varying environments: a two-level hierarchy, proceeding of the IEEE international conferece on robotics and automation, April 1988, pp. 1644-1649
60. Kayton.M (1989) Navigation: Land, Sea, Air, Space. IEEE Press,
61. Kim. D. H, (2009) Escaping Route Method for a Trap Situation in Local Path Planning International Journal of Control, Automation, and Systems (2009) (3):495-500 DOI 10.1007/s12555-009-0320-7
62. Kleeman. L, Kuc. R, (1995) Mobile robot sonar for target localization and classification, Int. J. Robotics Res. 14 , pp.295–318.
63. Komoriya, K. and Oyama, E., (1994), "Position Estimation of a Mobile Robot Using Optical Fiber Gyroscope (OFG)." International Conference on Intelligent Robots and Systems (IROS '94). München, Germany, September 12-16, pp. 143-149.
64. Konolige. K, Myers. K, Saffiotti. A and Ruspini. E, (1997) The Saphira architecture: a design for autonomy, Journal of Experimental and Theoretical Artificial Intelligence, pp. 215--235.
65. Korf, R.E, (1987) Real-time Heuristic Search: First Results, the Sixth National Conference on Artificial Intelligence.
66. Kuc. R and Siegel .M .W (1987) Physically based simulation model for acoustic sensor robot navigation. IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI-9(6), pp.766-778.
67. Kuc.R, (1990), A spatial sampling criterion for sonar obstacle detection. IEEE

Transactions on Pattern Analysis and Machine Intelligence, PAMI-12(7):686{690, July 1990.

68. Kyriakopoulos K.J, Saridis G..N., (1993), An Integrated Collision Prediction and Avoidance Scheme for Mobile Robots in Non-Stationary Environments, Automatica, Vol 29, 1993, NO. 2, pp. 309-322
69. Latombe, J.C. (1991), Robot Motion Planning, Kluwer Academic Publishers
70. Latombe, J.C. (1993). Robot motion planning. New York: Kluwer Academic.Laprie. J.C., (1985), Dependable Computing and Fault Tolerance: Concepts and Terminology. 15th International Symposium on Fault Tolerant Computing Systems, pp.2-11.
71. Laprie.J.C, (1985), Dependable Computing and Fault Tolerance: Concepts and Terminology. 15th International Symposium on Fault Tolerant Computing Systems, pp.2-11
72. Laprie.J.C, (1992), Dependability: Basci Concepts and Terminology – In English, French, German and Japanese. Vienna: Springer-Verlag.
73. Leonard. (1992) Directed Sonar Sensing For Mobile Robot Navigation, Kluwer Academic Publishers, USA
74. Linder, S. P. , (1998). Behavior Based Robust Fuzzy Control. In Proceedings of the 1998 IEEE ISIC/CIRA/ISAS Joint Conference, Gaithersburg, Maryland, USA, pp. 233-238.
75. Liu. Y, Yang. Y.R. (2003) .Reputation Propagation and Agreement in Wireless Ad Hoc Networks,. Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC).
76. Lozano-Perez.T, Wesley. M.A.,(1979), An algorithm for planning collision-free paths among polyhedral obstacles, Commun. ACM 22 (10),pp.560-570.
77. Lumelsky V.L, Stepanov A.A, (1987), “Path- Planning strategies for a point mobile automaton moving amidst unknown obstacles of arbitrary shape ”, algorthmica, 1987, 2, pp. 403-430
78. MacKenzie.P and Dudek.G, (1994) Precise positioning using model-based maps. Proceedings of the IEEE International Conference on Robotics and Automation, pp.1615-1621.

79. Martinelli. A, Siegwart. R, (2003) Estimating the Odometry Error of a Mobile Robot during Navigation- Proceedings of European Conference on Mobile Robots (ECMR03),

80. Mehyar. M, Spanos. D, Pongsajapan. J, Low. S.H, Murray. R.M, (2005) Distributed Averaging on Asynchronous Communication Networks, Proceedings of the Joint 44th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC'05), Seville, Spain.

81. Miyajima,K, Ralescu. A, (1994) “Spatial Organization in 2D Segmented Images: Representation and Recognition of Primitive Spatial Relations”, Fuzzy Sets and Systems, vol. 65, iss. 2/3, pp. 225-236.

82. Morgan A. V. and Russel P. I. (1998) What on Earth Archieves.

83. Nelson, V.P., and Carroll. B.D., (1982), Fault-Tolerant Computing (A Tutorial), presented at the AIAA Fault Tolerant Computing Workshop, Fort Worth, Tex., November 8-10.

84. Nico. D, Daprati. E (2009), The egocentric reference for visual exploration and orientation, Brain and Cognition 69 ,pp. 227- 235

85. Otte. M. W, Richardson. S.G, Mulligan.J, and Grudic. G, (2009) Path Planning in Image Space for Autonomous Robot Navigation in Unstructured Environments Journal of Field Robotics 26(2),pp. 212–240

86. Oskoei. M. A, Gan. J.Q, and Hu.H, (2009) Adaptive schemes applied to online SVM for BCI data classification, The 31st Annual Int. Conf. of the IEEE Engineering in Medicine & Biology Society, Hilton, Minneapolis, Minnesota, 2-6 September 2009.

87. Oskoei. M.A. and Hu.H, (2009) Adaptive myoelectric Human-Machine Interface for video games, Proc. of IEEE International Conference on Mechatronics and Automation, Changchun, 9-12 August 2009

88. Payton. D. W. Rosenblatt. J. K., and Keirsey D.M. 1990 Plan guided reaction, IEEE Trans. on Systems, Man, and Cybernetics 20 (6).

89. Perzanowski. D, Schultz. A, Adams. W and Marsh. E, (1999) “Goal Tracking in a Natural Language Interface: Towards Achieving Adjustable Autonomy”, In Proceedings of the 1999 IEEE International Symposium on Computational

Intelligence in Robotics and Automation, Monterey, CA, pp. 208-213.

90. Porrill. J, (1988) Optimal combination and constraints for geometrical sensor data, The International Journal of Robotics Research 7 (6), pp. 66-77
91. Pirzadeh, A., Snyder, W., (1990) A Unified Solution to Coverage and Search in Explored and Unexplored Terrains Using Indirect Control, IEEE International Conference on Robotics and Automation.
92. Pullum. L, (2001), Software Fault Tolerance Techniques and Implementation, Artech house, INC. 685 Canton ST. Norwood, MA 02062
93. Rau.A.V, Dunningham. J. A, Burnett. K. (2003) Measurement-Induced Relative-Position Localization Through Entanglement, Science 22:Vol. 301. no. 5636, pp. 1081 – 1084
94. Rencken, W. D., (1994), "Autonomous Sonar Navigation in Indoor, Unknown, and Unstructured Environments."1994 International Conference on Intelligent Robots and Systems (IROS '94). München, Germany, September 12-16, pp. 431-438.
95. Saffotti, A. (1998) Fuzzy logic in Autonomous Robot Navigation: A case study, ser. Handbook of Fuzzy Computation, Oxford Univ. Press and IOP Press, Oxford, UK.
96. Saffiotti. A, (2000) Fuzzy logic in autonomous navigation, in: D. Driankov, A. Saffiotti (Eds.), Fuzzy Logic Techniques for Autonomous Vehicle Navigation, Physica-Verlag, Heidelberg, New York, pp. 3–22.
97. Sajotti. A, Wesley. L, (1996) Perception-based self localization using fuzzy locations, in: M. van Lambalgen (Ed.), Reasoning with Uncertainty in Robotics, Lecture Notes in Computer Science, Springer Berlin, pp. 368–386.
98. Samperio. R,Hu. H,Gu. D, (2009), Implementation of A Localization-oriented HRI For Walking Robots In The Robocup Environment, International Journal of Information Acquisition Vol. 5, No. 4 (2008) 1–17.
99. Seraji. H, Howard. (2002) Behavior-based robot navigation on challenging terrain: A fuzzy logic approach, IEEE Transactions on Robotics and Automation 18 (3), pp. 308–321.
100. Sgorbissa, A, Arkin. R. C, (2003) Local navigation strategies for a team of robots. Robotica, 21, pp. 461-473.

101.Shibata. F, Ashida. M, Kakusho. K, Babaguchi. N, and Kitahashi. T, (1996) “Mobile Robot Navigation by User-Friendly Goal Specification”, In Proceedings of the 5th IEEE International Workshop on Robot and Human Communication, Tsukuba, Japan, pp. 439-444.

102.Spiers. A J, Warwick. K, Gasson. M N, Ruiz. V F (2007), "Issues impairing the Success of neural implant technology", Applied Bionics and Biomechanics, Vol.3, Issue.4, pp.297-304.

103.Stachniss. C, Burgard. W, (2003) ,Exploring Unknown Environments with Mobile Robots using Coverage Maps, University of Freiburg, Department of Computer Science, D-79110 Freiburg, Germany.

104.Stopp. E, Gapp. K.P, Herzog. G, Laengle. T and Lueth. T, (1994) “Utilizing Spatial Relations for Natural Language Access to an Autonomous Mobile Robot”, In Proceedings of the 18th German Annual Conference on Artificial Intelligence, Berlin, Germany, pp. 39-50.

105.Talluri, R., and Aggarwal, J., (1993), “Position Estimation Techniques for an Autonomous Mobile Robot - a Review.” in Handbook of Pattern Recognition and Computer Vision, World Scientific: Singapore, Chapter 4.4, pp. 769-801.

106.Tan.G, He H, Aaron. S, (2006) Global optimal path planning for mobile robot based on improved Dijkstra algorithm and ant system algorithm, Journal Of Central South University Of Technology,Vol.13 No.1, pp.80-86

107.Tompkins. P, (2005) Mission Directed Path Planning For Planetary Rover Exploration, The Robotics Institute Carnegie Mellon University Pittsburgh, PA 15213

108.Tonouchi, Y., Tsubouchi, T., and Arimoto, S., (1994), "Fusion of Dead-reckoning Positions With a Workspace Model for a Mobile Robot by Bayesian Inference." International Conference on Intelligent Robots and Systems (IROS '94). Munich, Germany, September 12-16, pp. 1347-1354.

109.Tsitsiklis. J. N. (1984) Problems in Decentralized Decision Making and Computation,. Ph.D. Thesis, Department of EECS, MIT.

110.Wang. M, Liu J.N.K. (2004) Online path searching for robot autonomous navigation, in: IEEE Conference on Robotics, Automation and Mechatronics, RAM, Singapore, pp. 746–751

111. Wang. M, Liu. J.N.K, (2008) Fuzzy logic-based real-time robot navigation in unknown environment with dead ends, Robotics and Autonomous Systems 56 ,pp.625–643

112. Warwick. K, Nasuto. S J (2007), "Rational AI: What does it mean for a machine to be intelligent?", IEEE Instrumentation and Measurement Magazine, Vol.9, Issue.6, pp.20-26.

113. Warwick.k (2007), "The Promise and Threat of Modern Cybernetics", Southern Medical Journal, Vol.100, Issue.1, pp.112-115.

114. Washington. R, Golden. K, Bresina. J, Smith. D.E, Anderson. C, Smith. T, (1999) Autonomous Rovers for Mars Exploration NASA Ames Research Center, MS 269-2 Moffett Field, CA 94035 650-604-5000.

115. Weng.L, Song,D. Y. (2005) Path planning and path tracking control of unmanned ground vehicles (UGVs) Dept. of Electr. Eng., North Carolina A&T State Univ., USA; System Theory. SSST '05. Proceedings of the Thirty-Seventh Southeastern Symposium on,pp. 262- 266

116. Wijk.O, Christensen.H.I., (2000) Localization and navigation of a mobile robot using natural point landmarks extracted from sonar data, Robotics Autonomous Systems 31, pp.31–42.

117. Ye. C, Wang. D, (2000) A novel behavior fusion method for the navigation of mobile robots, in: IEEE International Conference on SMC, vol. 5, pp. 3526–3531

118. Yeap.W.K., (1988) Towards a computational theory of cognitive maps. Artificial Intelligence, 34, 1988, pp. 297-360.

119. Yeap.W.K. and Jefferies. M.E., (1999) Computing a representation of the local environment. Artificial Intelligence, 107, pp. 265-301.

120. Yen J., Pfluger N., (1995), A Fuzzy Logic Based Extension to Payton and Rosenblattt’s Command Fusion Method for Mobile Robots, IEEE Transactions on Systems, Man, and Cybernetics, Vol 25, No 6, June 1995, pages 971-978

# Appendix

1. simulation environment se.wld file

*width 10000*
*height 10000*
*50 0 50 3600 ;corridor bottom left wall*
*1650 0 1650 2000 ;corridor bottom right wall*
*50 3600 4050 3600 ; corridor top wall*
*1650 2000 6050 2000;*
*6050 2000 6050 3600;*
*3050 5400 3450 5400;    center square*
*3050 5800 3050 5400;*
*3050 5800 3450 5800;*
*3450 5800 3450 5400;*
*4150 5900 4750 5900; center triangle*
*4150 5900 4350 6200;*
*4350 6200 4750 5900;*
*5250 5800 6050 5800; center rectangle(horizontal)*
*6050 5800 6050 6000;*
*6050 6000 5250 6000;*
*5250 6000 5250 5800;*
*6550 5600 7050 5600; center rectangle(vertical)*
*6550 5600 6550 6600;*
*7050 5600 7050 6600;*
*6550 6600 7050 6600;*
*10000 3600 8700 3600; right rectangle by the wall middle*
*8700 3600 8700 5100;*
*8700 5100 10000 5100;*
*10000 5900 9500 5900;right rectangle by the wall top*
*9500 5900 9500 5400;*
*10000 5400 9500 5400;*
*10000 1500 9000 1500;*
*9000 2900 10000 2900;*
*9000 1500 9000 2900;*
*10000 500 9500 500; bottom corner*

*9500 500 9500 0;*
*9000 500 9000 0;    bottom rectangle*
*9000 500 8200 500;*
*8200 500 8200 0;*
*7600 600 7800 0; bottom triangle*
*7600 600 7400 0;*
*6600 600 6600 0; bottom polygon*
*6600 600 6000 600;*
*6000 600 5600 0;*
*1200 8000 0 8000; top left rectangle*
*1200 8000 1200 6200;*
*0 6200 1200 6200;*
*0 5350 250 5600;   top left diamond*
*250 5600 500 5350;*
*500 5350 250 5100;*
*250 5100 0 5350;*
*3350 6200 4000 6200; center bigger square*
*3350 6850 3350 6200;*
*3350 6850 4000 6850;*
*4000 6850 4000 6200;*
*3500 7100 3500 7600; center polygon*
*3500 7100 4000 7100;*
*3500 7600 3750 8000;*
*3750 8000 4000 7600;*
*4000 7600 4000 7100;*

*position 850 500 90*