ISSN 0280–5316
ISRN LUTFD2/TFRT--5545--SE

Collision Avoidance for Autonomous Underwater Vehicles

Helena Flygare

Department of Automatic Control
Lund Institute of Technology
December 1995

Department of Automatic Control
Lund Institute of Technology
P.O. Box 118
S-221 00 Lund Sweden

Document name
MASTER THESIS

Date of issue
December 1995

Document Number
ISRN LUTFD2/TFRT--5545--SE

Author(s)
Helena Flygare

Supervisor
Rolf Johansson (LTH), Ola Dahl (Kockums)

Sponsoring organisation

Title and subtitle

Collision Avoidance for Autonomous Underwater Vehicles

Abstract

This Master Thesis project is a pilot study in developing a collision avoidance function for an autonomous underwater vehicle, also called AUV. A collision avoidance function is necessary to ensure safe operation of an AUV. The AUV gets information about the surroundings through an active sonar. The image from the sonar has to be processed to extract information about the environment, e.g., detection of obstacles. If there are any obstacles in the AUV's way, the collision avoidance function has to make decisions to elude a collision.

A literature study of sonar system, path planning and collision avoidance functions has been done. A summary of the literature study is presented. A design approach for collision avoidance in an AUV is given. To minimise the energy consumption and the risk of a collision, a three level operation is suggested. The first level plans a global path, the second level plans a local path through the obstacles in the sonar image, and the third level generates a pure reflexive action to avoid a collision with unforeseen obstacles.

A simulation environment is built up containing the AUV dynamics, an autopilot, a simple sonar model and an implementation of a collision avoidance function of the third level.

Key words
autonomous underwater vehicle, AUV, collision avoidance, path planning, sonar.

Classification system and/or index terms (if any)

Supplementary bibliographical information

ISSN and key title
0280–5316

ISBN

Language
English

Number of pages
43

Recipient's notes

Security classification

The report may be ordered from the Department of Automatic Control or borrowed through the University Library 2, Box 1010, S-221 03 Lund, Sweden, Fax +46 46 110019, Telex: 33248 lubbis lund.

Table of contents

1 Introduction

This Master Thesis project is a pilot study in developing a collision avoidance function for an autonomous underwater vehicle (AUV). The work has been done at Kockums Submarine Systems Ltd. in co-operation with the Department of Automatic Control at Lund Institute of Technology.
This thesis was supervised and supported by

- Dr Ola Dahl, Software Development, Kockums Submarine Systems Ltd.
- Dr Rolf Johansson, Department of Automatic Control, Lund Institute of Technology

1.1 Task definition for the MSc Thesis Project

1.1.1 Collision avoidance for an autonomous underwater vehicle

The MSc Thesis project is considered to be a pre-study to a possible autonomous underwater vehicle (AUV) project at Kockums Submarine Systems Ltd. The goal is to produce a collision avoidance function for an AUV. A collision avoidance function shall automatically plan and perform an evasive action. The purpose is to avoid obstacles in the surroundings. Conceivable part functions are:

1. Measuring of position with e.g. active sonar, GPS, DGPS or TN-systems.
2. Planning of a collision free path. The method can vary from easy heuristic algorithms to more arduous computations e.g. minimal time, minimal energy, etc.
3. Performing of the motion. This means automatic operation control in Cartesian co-ordinates (x,y,z).

The aim is a collision avoidance function containing these three part functions. The function is to be implemented as a prototype system that can be demonstrated. The implementation shall be done in Matlab/Simulink.

1.1.2 Operational sub goals

The following sub goals are formulated:

- Literature study of published articles
- Study of sensors and contact with suppliers
- Algorithm for a path planner
- Integration of path planning and operation control
- Simulation model in Simulink
- Simulation and visualisation in a prototype system

1.2 Outline

Chapter 2 gives a general view of problems associated with the operation of an AUV. It also states the requirements on the vehicle and the environments for its use. A design approach for a collision avoidance function is suggested in chapter 3. The performance of the collision avoidance function depends strongly on the sensors, i.e. the sonar system, and the processing of the acquired information. This is discussed in chapter 4. A summary of a literature study of what has been elaborated so far in the domain of collision avoidance functions and path planning algorithms is presented in chapter 5. Chapter 6 describes a design of path planning and collision avoidance function made by the author. A reactive collision avoidance function in two and three dimensions is also presented in this chapter. The simulation environment is presented in chapter 7. An evaluation of the vehicle's dynamics and autopilot is also given in this chapter, together with a simple sonar model. Some simulation examples using the complete model, i.e. the sonar model, the implementation of the collision avoidance function, obstacles, the AUV model and the autopilot are given in chapter 8. A concluding discussion about the collision avoidance function and the sonar for use in an AUV is given in chapter 9.

2 General aspects and difficulties with an AUV

2.1 What need do we have of an autonomous underwater vehicle?

A remotely operated vehicle (ROV) is a submersible vehicle that gets its power and control signals through a cable, also called umbilical cord. The ROV has been developed and used for many years, see e.g. [7], where a Norwegian ROV is described. Through the cable the ROV transmits video signals from a CCD camera, or an active sonar placed on board. A person ashore or on a ship steer the vehicle by a joystick with the help of the transmitted images. However, the ROV has a limited range of operation due to the cable. This has led to the wish of making a vehicle without the cable and this is the reason for creating an AUV. Since an AUV doesn't have an umbilical cord, one has to ensure that it can behave in an "intelligent" way, with enough power to return to the base, when the mission is accomplished. This arises a great number of difficulties, such as power supply, path planning, guidance, navigation, control, communication, sensors and many more. The technological advances in the areas of power supplies, computer engineering, automatic control and communication, have made it possible to develop autonomous underwater vehicles. Though interesting, many of the difficulties associated with the AUV lay outside the scope of this thesis. The AUV can be used for purposes beyond reach of traditional methods, such as the ROV and ordinary submarines. There are many applications for an AUV both in the civilian and the military area. Some examples of civilian applications are: pipeline inspection, taking of specimens of the water, control of the sea environment, sea bed mapping and laying of cables offshore. Military utilisation can be: advanced reconnaissance, piloting a submarine or a vessel through a mine field, mining, diverting attention from the manned submarine etc.

2.1.1 Navigation

The navigation system in an AUV is closely related to the collision avoidance function, as the AUV needs an accurate knowledge of its position, orientation and velocity, but also of the objects surrounding it. Navigation is one of the areas that requires the most development, in order to minimise the position inaccuracy. Important research areas are e.g. improved knowledge of the position, refined acquisition of data from sensors and energy storage. The navigation can be either relative or absolute. The difference of the two is stated below.

2.1.1.1 Relative navigation

Measurement of the position, the velocity or the acceleration can be obtained from inertial navigation systems, Doppler log sonar, correlation sonar, laser scanning or by measuring the change of the Earth's magnetic field. The problem these methods have in common is that they tend to drift, i.e. the inaccuracy increases with time. The position and the velocity are only relative to the surroundings. The position and the velocity are sometimes given by integration of the acceleration, which increases the error further. Some kind of position fixes are needed after a while.

2.1.1.2 Absolute navigation

The most accurate navigation information currently available, at least to the authors knowledge, is probably the Global Positioning System (GPS) [18], where the position is broadcasted through satellites. GPS gives accurate position measurements, but for an AUV this is possible only when the antenna is out of the water, because electromagnetic waves propagate only a few metres in water. This gives rise to another problem: the AUV has to surface to get the position fixes. In doing so, it loses a lot of energy, at least when it is submerged at large depth. Another problem is that when surfaced, the water washes over the antenna which gives problem with the receiving of the signals. GPS has two levels of accuracy, the Standard Positioning Service (SPS), and the Precise Positioning Service (PPS). The main difference of the two is that the PPS uses cryptographic keys to decode and remove error in broadcasted signals, which yields a higher accuracy than the SPS. The PPS is classified for the American military and has an accuracy of 18 metres. The SPS is for civilian users, and is intentionally disturbed with Selective Availability (SA) to degrade the accuracy to 100 metres. The precision acquired with the SPS can be increased by differential methods, called differential GPS (DGPS), that can improve the accuracy to be better than the PPS. Differential GPS results in an accuracy of 5-10 metres.

If the AUV is to operate in a given area, acoustic, magnetic or visual beacons can be positioned at predetermined points on the bottom of the sea. When the AUV operates in the vicinity of these, it can update its position knowing the exact location of the beacon. The main problem of these methods is that the beacons need power supply. This problem can be eliminated in the case when magnetic beacons are utilised, by using permanent magnets. However, for general missions this is not a very attractive way of getting position fixes, on account of heavy expenses in setting out the equipment and limitations of the operation area.

Another method is learning to recognise natural features, which are either unique or characteristic for a particular position. This can be done using a multibeam sonar to measure variations of depth at different angles in bottom backscatter, or when working close to the sea bed, a CCD camera can be used to recognise features and comparing them with a detailed nautical reference chart. This approach doesn't give a unique global position for every measurement, since similar features can occur at different locations. By including a rough estimate of the velocity the position error can be greatly reduced, see [2]. This method can not be used when the sea bed is flat and lacking structure.

Yet another method is based on variations in the Earth's gravity field, which are matched against a stored gravity gradient map [12].

2.2 Stated requirements on the AUV

The following technical specification is made on the AUV for a possible project at Kockums Submarine Systems Ltd.

- Endurance of 96 hours
- Operating depth of 1000 metres

- Transit speed of 8 knots
- Navigational accuracy of 50 metres per hour of travel
- Stirling motor 15 kW
- Weight 5-10 metric ton

The AUV is not equipped with thrusters, i.e. it can not change the position upwards or downwards without forward velocity. The AUV is to operate close to the shore, as well as at great depths and close to the sea bed. It has a geometry of an ordinary submarine except for the tower (actually more like a torpedo). In some areas, digital nautical charts are available. A typical resolution is 12-25 metres in the skerries.

As a comparison, some performance data are given for an already built AUV, Theseus [6] from Canada.

Theseus has the following specification

- A range of 250 nautical miles
- A working depth of 3.250 feet
- A cruising speed of 4.0 knots
- A payload bay of 96 inches long by 44 inches inside diameter

Another AUV built under the Marine Science and Technology Programme (MAST) of the European Union, is Martin from Denmark, Portugal and France [1],[3].

3 A design approach

A design approach for a collision avoidance function is suggested. Each of the blocks in Fig. 3.1. below will be explained.

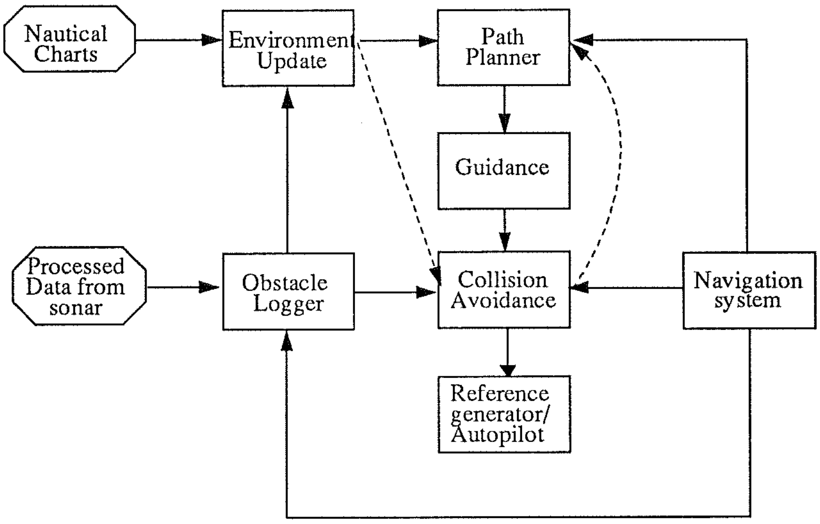


Fig. 3.1 A design approach

Navigation system: The position and the velocity of the AUV, acquired by methods described in chapter 2, are delivered to the collision avoidance block, the obstacle logger and the path planner.

Nautical Charts: Available nautical charts are loaded down in the AUV before the mission starts.

Environment Update: The environment is updated based on what the AUV has seen during the journey. Information about the detected obstacles is delivered from the obstacle logger.

Obstacle logger: The obstacles position and size, obtained from the processed data from the sonar, are logged. The navigation system gives the AUV's position and velocity, which gives the obstacle logger a reference for determination of the obstacles position. To get more precise values of the obstacles position and size, a Kalman filter can be used, see e.g. [4].

Path Planner: An optimal global path (e.g. minimum energy or minimum distance) is planned based on currently available information about the environment. The global path can be planned at different time intervals, and/or at request from the collision avoidance function.

Guidance: The planned path is divided into way points that are delivered to the collision avoidance block, i.e. the trajectory is planned given the curve from the path planner.

Collision avoidance: To avoid unforeseen obstacles along the path, the collision avoidance function should be active between the way points. Depending on the circumstances, it could be a pure evasive impulse, or a more solid function that plans a new path when an obstacle is encountered. If the former is used, the path planner should be active more often to minimise the energy consumption. It's important that the collision avoidance function contains a pure reflexive action. The reason for this is that even if a collision-free path is planned, e.g. based on a sonar image showing the obstacles, new obstacles can be detected along the path, e.g. due to uncertainties in a sonar image. If the collision avoidance function is to plan a path between the obstacles the suggestion is that the collision avoidance function should operate at two levels. At the higher level a new local path between the obstacles should be planned and at the lower level a pure evasive action is active for obstacles that haven't been detected during the planning phase. A new logic block is then needed to decide which level is to set the new bearing. The reflexive-type of obstacle avoidance should always have the highest priority. This technique is used in Martin [1],[3], where a local path based on information from a sonar image, is planned every seven seconds. This path may be overruled by a Reactive Avoidance Controller, when an obstacle is coming too close.

Processed data from sonar: The data from the sonar should be processed (signal processing as well as image processing) to extract more accurate information from the sonar data.

Reference generator/Autopilot: Controllers to keep the desired depth and the desired course.

4 Sonar systems

The tactile organs of the AUV are the sensors. From these it gets the main information about the environment. As there isn't a human operator on board taking decisions, the signals from the sensors have to be processed automatically, which isn't a trivial problem. To extract most information from the environment an active sonar should be used. The sonar has to be well adapted to its purpose, e.g. if the AUV should operate close to the shore or in deep sea. Some sonars can't take the heavy pressure when the AUV is submerged at great depth and others aren't efficient when operating near the coast.

The raw data from the sonar has to be processed before it can be used. For collision avoidance, the most important processing tasks are object recognition and classification.

4.1 How a sonar works

An active sonar transmits sound, which is reflected when hitting objects that lay in the way of the sound wave. The echo is then received by the sonar. The transmission is done by a projector, which transforms electricity to ultrasonic sound. A hydrophone transforms the ultrasonic sound to electricity. The receiver is often an array of hydrophones. A hydrophone is a stave with a preamplifier. There are different configurations of the array of hydrophones [15]. In an AUV the most appropriate configuration is a circular arc or a line array. These configurations are suitable for a forward looking sonar. Several hydrophones work interactively to form a beam in a given direction and with a given width. Thus, when dealing with the returning sound, the hydrophones form beams in the directions that the sonar is "listening" to. The beams are formed in all directions of the field of view. The field of view is defined by the range of the sonar, a horizontal and a vertical sector. Fig. 4.1. shows the field of view for two dimensions, where α defines the horizontal angle. The vertical sector is defined as the corresponding angle relative the horizontal reference. The procedure of transmitting sound is the same as when receiving, i.e. the transmitted sound is formed as beams by the projectors. The beams cover up the whole field of view.

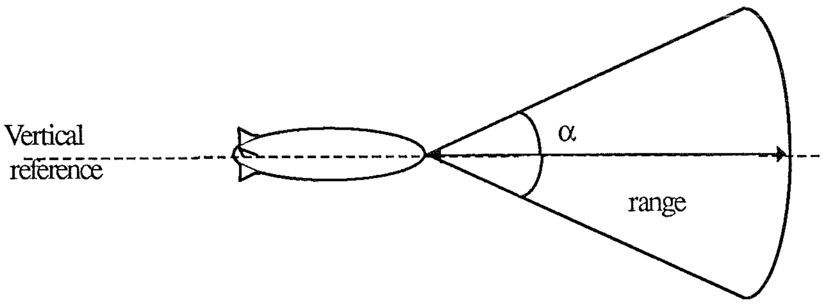


Fig. 4.1 The field of view of the sonar. α defines the horizontal sector. The corresponding angle with the horizontal reference defines the vertical sector.

The electric signals from the hydrophones can be grouped together to form an image. The image is an array of pixels. The image could be in colour or black and white. For a black and white image, each pixel has a value that corresponds to the value of the energy in the signal. If the pixel values are normalised, zero means the colour is black and the value one means white. Values in between are grey tones. If only the horizontal sector is scanned, the image has the same geometry as the field of view in Fig. 4.1. If the vertical sector is also scanned, the image will be three dimensional. If an object is moving, then the sonar is able to decide the objects velocity through the Doppler shift. It is then possible to extract information such as size, position and velocity of a detected object from the signals of the sonar.

For a large horizontal sector of 120° or more the circular arc configuration is best suited, and for sectors of 90° or less the line array is best [15]. The number of staves varies depending on which configuration is used, and also on the width of the formed beam. A smaller beam width requires a larger amount of staves, and vice versa. The beams of a circular arc have all the same spacing and uniform width, whilst with a line-array the spacing and the width of the beams are smaller straight forward (higher resolution) and larger at the edges of the field of view. This has to be considered when the image is processed. The formed beam has side lobes as sound is a wave form. The magnitude of the side lobes depends on the system. The effect of the side lobes will be discussed in the next section.

The resolution of the sonar depends on :

- The width of the beam, a narrow beam gives a high angular resolution
- Pulse length, a small pulse length gives a high resolution
- Range, the resolution decreases with larger range
- Side lobe levels for the transmitted beam as well as for the received beam
- Minimum detectable signal level
- Receiver noise level

Vendors of sonar systems usually don't specify the last three items of the list above. They are however very important when choosing a sonar system for an AUV, and should be asked for.

4.2 Difficulties associated with a sonar

There are several difficulties associated with ultrasonic measurement [19]. A couple of them will be explained and clarified.

As sound is a wave form it reflects when hitting the boundary of media with different acoustic impedance. This is exactly the purpose of the active sonar, but into the bargain comes unwanted side-effects. The principal one is *reverberation*, which is reradiation of acoustic energy transmitted by the sonar. Reverberation originates in that the water is inhomogenous, that is the sea contains tiny dust particles (that make the water appear blue), algae, fish, seabed structure etc. This causes variations of the acoustic impedance in the water which in turn cause reradiation of a part of the acoustic energy. This phenomenon is called scattering and the reverberation is the total contribution from all scatterers. Reverberation is often the principal limitation on a system's performance. The reverberation can be decreased by modifying the beam width and pulse length. Another problem caused by variations in the acoustic impedance is the boundary e.g. between cold and warm water or between water with varying salinity. This may lead to erroneous interpretation of the sonar image.

Yet another problem is *multiple detection,* which derives its origin from that a pulse can reflect several times between the object and the transducer. This may lead to detection of a "ghost object" behind the real object. The direction of the reflecting wave depends on the structure of the object's surface. Due to this, the reflecting wave may never reach the hydrophone, which in turn leads to erroneous interpretation of the object's size.

The side lobe levels may also create a problem. If they are high, and a large and a small object are in the field of view in different directions, the echoes from the small object can be swamped by echoes from the large object, received through the side lobes, leading to the smaller object remains unseen , see [15].

4.3 Image processing of a sonar image

The image from a sonar is three dimensional and if time is considered, it is four dimensional, as information about the velocity of an object can be given by Doppler shift. The image from a sonar is very noise corrupted. Either a filtering of noise is needed before the object detection is done, or an object detection, which is insensitive to noise, has to be used.

Except going through noise reduction, the image has to be further processed and a classification of obstacles has to be done. There are different ways of filtering and analysing an image, see e.g. [8]. Some examples are:

- grey tone analysis, which modifies the image by analysing and changing the grey tones in the pixels, e.g. to increase the contrast or to detect edges.
- image restoration, where e.g. the geometrical effects of the formed beams are eliminated from the signal, e.g. compensation of varying beam width in a line array sonar.
- morphological analysis, that extracts image components that are useful in description and representation of region shape, such as boundaries, skeletons and the convex hull.
- structural analysis, which recognise patterns from structural relationships inherent in a patterns shape, e.g. sea bed structure.

The easiest way to perform detection and classification of objects in the image is probably by thresholding, which is a form of grey tone analysis. Thresholding means that the value of every pixel is compared with a threshold value. If the value of the pixel is higher than the threshold, then the value of the pixel is set to the maximum value, otherwise it is set to the minimum value. This gives a binary image. Adjacent pixels that have the same value are considered to belong to the same object. Before thresholding the image, it's important that the grey tones are equally distributed in the image. If this is not the case, i.e. if the signal levels are not equally distributed between the minimum and the maximum level (which they often aren't), then the corresponding image may tend to the darker or the brighter direction, which can make it difficult to get a good threshold value. The remedy can be equalisation by histogram, see [8]. The method of thresholding an image have many drawbacks. The echo from an object depends e.g. on the surface of the object and the inclination of the beam, which in turn affects the values of the pixels representing the object, which can lead to erroneous interpretation of the object’s size. Objects that are small can remain undetected, due to low signal levels if the threshold value is set too high. By using more than one threshold value this problem decreases. The resulting image will then be in different grey tones and not in black and white. The threshold settings should be carefully chosen. If a threshold value is too high, one can risk that an object remains undetected and if it's too low it leads to false alarms. The threshold value should be changed adaptively so that the processing of the image becomes insensitive to variations of the background level. The background level depends for instance on the reverberation. The reverberation changes with variations in e.g. temperature, homogeneity or salinity of the water, and should therefore be taken under consideration when the image is analysed. An example of a sonar image processing in an AUV application is given in [9].

Expert systems or neural networks can also be used to process data and perform object recognition and classification.

5 A summary of a literature study of path planning and collision avoidance functions

Real-time mission planning and obstacle avoidance are very time critical concerning the AUV. The solution of making an obstacle avoidance function that works generally for both moving and non-moving objects has not yet been seen by the author. Moving obstacles are harder to deal with, as their direction and speed can be arbitrary and they behave in an unpredictable way. The work that has been done for fix objects can be grouped in three approaches : artificial potential fields, neural networks and space model methods. A brief description of these approaches is given in section 5.1, 5.2 and 5.3. The potential fields and the space model approach have been directly applied from robotics, in some cases without further reflections of what is appropriate for the particular use of the AUV. Dealing with moving obstacles the work haven't been so methodical. Some approaches works better than others, but more research has to be done in the area.

The navigation of an AUV through an environment with obstacles can generally be decomposed into three separate steps:

Path planning: A collision-free curve connecting the AUV's initial and final positions is constructed given geometric data and desired destination. No considerations are taken to the dynamics of the vehicle.

Guidance: Given the curve from the path-planner the trajectory planning is solved. A time parametrization for the curve, subject to certain constraints, is elaborated.

A **control** scheme is devised to make the physical AUV follow the reference trajectory as close as possible.

The methods described here either uses these steps as they are or integrate them.

To make the problem easier the AUV is reduced to a point and the obstacles are grown with the radius of safety of the AUV. The radius of safety is defined as the radius of the minimum circle around the AUV, which guarantees that the AUV can manoeuvre safely without colliding with an obstacle it hasn't seen yet. Most often the obstacles are represented as a point with an uncertainty circle around it. The following approaches are explained in two dimensions but can be extended in three dimensions.

5.1 Artificial Potential Fields

This approach has its origin in robotics research. It was first introduced by Khatib, in his 1980 Ph.D. dissertation [13]. He proposed a closed-form expression for a potential function and suggested to use its negative gradient as torque input. However, this method suffers from several problems, the most important one being undesired local minima. This will be explained in section 5.1.2. The most common use of a potential function has been for local path planning but have been considered useful for collision avoidance too. The obstacles are assumed to carry artificial electric charges, which give rise to a potential field around it. A collision between the AUV and the obstacle is avoided by a repulsive force between them, which is simply the negative gradient of the potential field. The goal position is assigned with a negative potential field, which attracts the AUV. A path planning can be done, where the path is selected from the minimum potential valleys, such that an estimate of the path length and the chance of collision is minimised. The changes of the AUV's position and orientation can be made smoothly and continuously as the shape and distance to the obstacle is well indicated by a continuous potential field.

5.1.1 The potential field function

There are many choices of potential function. The function must have an analytical expression for the shape of the obstacle. This analytical expression is one of the limitations of the potential field approach. There have been several attempts more or less successful to overcome this problem. See e.g. [14] that has come around the problem through a panel method.

5.1.2 Local minima

Getting trapped in a local minima is the main difficulty and is due to the path is planned towards lower potential. This results in that the AUV may reach a state of equilibrium or a potential basin and get caught. Potential fields are efficient concerning collision avoidance but not goal achievement if nothing is done about getting caught in a local minima. Several methods for getting around this problem have been suggested, but all of them brings on other problems, such as e.g. longer computation time. Some suggestions are:

1. Find a trial path and let the whole path be modified under the influence of the potential field [20]. Then there will be only a global minima (the goal) and the trapping in local minima can be avoided through optimisation techniques.
2. Extend the data structure that describes the detected obstacles with a visit count that keep an eye on how many times the AUV has been in the vicinity of a given obstacle [21]. The visit count is then multiplied by a gain and added to the potential function. The multiplication by a gain is to increase or decrease the sensitivity to go towards a detected object on several occasions.
3. Harmonic functions can be used to build an artificial potential field [14]. These have the property of not having any local extrema in a space free from singularities. A harmonic function is the solution to the Laplace equation,

$$\nabla^2 \phi = 0$$

   where ϕ is a scalar velocity potential.

5.1.3 The solution of the potential field function

The sum of the potentials and the AUV's kinetic energy can be placed in a Lagrangian formulation to decide the equations of motion for the AUV. These can be solved to decide a free path without obstacles to the goal.

5.2 Neural Networks

Neural networks have been very successfully used in image processing and in signal processing in general. The field for their use is expanding rapidly.

5.2.1 Background

A neural network is constructed as a multitude of non-linear computing elements (called neurons) organised as networks, reminiscent of the way which neurons are believed to be interconnected in the brain. Each neuron make a calculation of weighted inputs, where each weight (also called synaptic weight) can be chosen adaptively. The network is generally constituted of multi-layered neurons, i.e. it consists of layers of structurally identical computing nodes arranged so that the output of every neuron in one layer feeds into the input of every neuron in the subsequent layer.

5.2.2 Training by back propagation

The objective is to develop a training rule that adjusts the weights in each of the layer in a way that an error function is minimised. The error function, e, is the total squared error between the desired responses, d_q, and the corresponding actual responses, a_q, of nodes in the output layer

$$e = \tfrac{1}{2}\sum_{q=1}^{N}(d_q - a_q)^2$$

where N is the number of nodes in the output layer. To achieve the stated result the weights are adjusted in proportion to the partial derivative of the error with respect to the actual responses, a_q.

5.2.3 Neural networks applied on obstacle avoidance

Generally the inputs to the neural nets consist of the beam outputs from the forward looking sonar together with the position, the desired and the current speed, and the desired and the current course [17]. The neural nets integrates the three steps stated in the beginning of this chapter. Consequently the output is the rudder angles and propulsion power. Though, the steps are integrated, one can discern the steps within the nets. The dynamic model of the vehicle, and a geometric model of the vehicle can also be represented in the neural nets.

5.3 Space model methods

There are quite a few approaches of this kind but they can be widely organised in two groups: grid based and network/graph methods. In the network/graph approach the typical feature is that the environment is represented as a network of free space, which means a network of rectangles/quadrates in two dimensions, or boxes in three dimensions that doesn't contain any obstacles, or as a graph of obstacle vertices. This technique demand accurate sensor information, which is hard to get in practice, but offer elegant solutions.

As network/graph methods set a great requirement on the sensors they will not be discussed further, due to uncertainties of the active sonar. This section will only treat grid based methods.

The AUV's working space is divided in rectangles/quadrates in two dimensions or boxes/cubes in three dimensions of a predetermined length, width and in the case of three dimensions, height to define certain and uncertain regions of the environment. A certain region is an area known to be safe i.e. without obstacles whereas an uncertain one can mean that the area is unexplored or unsafe.

5.3.1 Dynamic programming

Dynamic programming is a well establish technique for optimisation. It will be explained how it is used in connection with AUV.

The procedure starts from the goal node and works back to the node where the AUV is located. Each element (rectangle/box) is assigned with a state, a cost, and a direction [11]. The state is the likelihood that the element is safe, the cost is the minimum distance needed from the start node to the goal node passing through the specific node (most often the Euclidean distance) and the direction points towards the contiguous node that lays on the optimal path. By definition the optimal path is the shortest safe trajectory from the start node to the goal node passing through a given node n.

The optimal path is sought by node expansion. Expansion of a node means that a cost and a direction are assigned to adjacent nodes. The expanded node is referred as the father node and the adjacent nodes as the sons. Only nodes that are sons can be expanded. Once they have been expanded they can't be expanded again.
The total cost can be expressed as:

$f(n)=g(n)+h(n)$

where $g(n)$ is calculated for each expansion, being the sum of the expanded node's $g(n)$ and the incremental cost of moving to node n, and $h(n)$ is the cost from node n to the vehicle's node, as this is unknown it has to be estimated.

There are different ways of expanding nodes. The most popular in robotics and for AUV use is A* search, as it is optimal in the way that it minimises the total distance from the start node to the goal node. A second one will also described, breadth-first search, in order to illustrate the hardships of A* search and dynamic programming in general.

5.3.1.1 A* search

The son node with the lowest overall cost is always expanded first. To make this possible the procedure has to have a list sorted by increasing overall cost. A new son node is sorted in the list and an expanded node is taken out of the list. More computation time is therefore needed to achieve the optimal solution. If the estimate of $h(n)$ doesn't exceed the true value, then A* is guaranteed to find the optimal path [11].

5.3.1.2 Breadth-first search

The node graph is searched a layer at the time. The goal node is first expanded. Then all its children in the same layer as the goal node. The method tries to expand as many nodes as possible within that layer before it proceeds with the next layer and stops when the node where the vehicle is, is reached. Sometimes the assignment is unsuccessful, for example when none of the safe nodes contiguous to the son node has been assigned with a cost. Something that remembers these nodes is therefore necessary, because when a new cost is available to a contiguous node (from a higher layer), the procedure has to reconsider these nodes for a new assignment. If this is not done then not so many safe nodes are assigned. The breadth-first algorithm minimises the number of layers being crossed and not the total distance.

5.4 An approach for moving obstacles

The main problem with moving obstacles is to keep track of them. This is particularly harder when there are many obstacles, moving in different directions and with varying velocities. An evasive action from one obstacle should not lead to the probability that a collision arises with another one.

One approach to solve the problem of moving obstacles is suggested in [4]. There it is assumed that the bearing of each obstacle is constant over a given period of time. By this assumption a trajectory of the obstacle can be calculated. The trajectory of the AUV is also calculated. If there is a crossing of the trajectories at the same instant then a collision is predicted. The first remedy of avoiding a collision is to change the AUV's velocity. If a collision is still predicted then the AUV changes bearing.

5.5 Evaluation of the methods

Each of the methods described in sections 5.1, 5.2 and 5.3 are powerful for planning a path. All methods can be used in the path planner block in chapter 3. For the collision avoidance function the situation is somewhat different. Suppose for example that one of the space model methods is used for finding a free path through the obstacles that are seen and have been seen by the sonar. As being said before the environment is represented with a grid. The size of the squares/boxes of the grid has to be chosen. If the size is too large, then even if the probability is small of finding an obstacle in a square/box, there is still a possibility of colliding if the AUV is to enter that square/box. If on the other hand the size is too small, then due to many squares/boxes have to be searched to find an optimal path, the calculation of a free path can take so long that a collision may occur before the path have been calculated. Another fact that has to be taken under consideration when choosing the size, is the resolution of the sonar. There is no point of making the squares/boxes small if the resolution is low. Another problem is the uncertainty in detecting an object, owing to the beam forming of the sonar as well as to the processing of its data. This uncertainty increases with the object's distance to the AUV. This means that a small obstacle can remain undetected until it is very close to the AUV. Another fact to be considered is the uncertainty (which also depends on, inter alia the resolution of the sonar) in the predicted position of an obstacle. This necessitates the use for a pure evasive impulse of the AUV to guarantee the exclusion of a collision. However, it's desirable that each manoeuvre consumes as little energy as possible and therefore the collision avoidance algorithm/path planner should be able to plan a path based on the information of the sonar too. The potential field method is expected to work better at this level, because it's consuming less time than methods that uses dynamic programming. However, the potential field approach is not as solid as the space model methods. One of the problems the potential field method have, is the detoriated efficiency of goal achievement, due to the problem of getting trapped in a local minima. Though, it has the great advantage of having a fast algorithm, which is very important in the use of the AUV. It is also easy to expand to higher dimensions. A potential for the neural networks is also seen.

6 A collision avoidance function

The study of path planning algorithms and sonar systems together with the AUV's dynamics has resulted in the following approach for the path planning and collision avoidance function.

6.1 A design of a collision avoidance function

For a good path planner the obstacles seen by the sonar should be logged and the environment should be updated, so that a complete picture of the surroundings should be taken into account when the path is planned. This will minimise the risk of a collision.

As the AUV's position changes, the detected object's position in the sonar image changes also. The obstacle log block described in chapter 3, should therefore keep track of the obstacles from frame to frame. The obstacle logger should also keep track of all the obstacles positions and estimate if a detected obstacle is new, or if it has already been detected. The position error in the navigation system may create a problem. This error propagates to the position for the detected obstacle, which may lead to hardship in determining if the obstacle is a new one or if it has already been detected. If it hasn't been detected yet, the obstacle is logged and the environment can be updated. If the AUV doesn't get position fixes as explained in section 2.1.1, the navigation error gets larger with time, which increases the variance of a detected obstacle's position. Based on the logged obstacles and the nautical charts the path planner could plan an optimal path. However, all the sources of uncertainties, such as e.g. navigation error, detection of obstacles, necessitates the use of a reactive obstacle avoidance function.

To minimise the risk of collision and the energy consumption the collision avoidance function should contain three levels.

6.1.1 The first level

At this level there's a solid path planner that plans an optimal global path, with respect to the energy consumption. All information available about the surroundings, such as nautical charts and logged obstacles, should be taken under consideration. An optimal path should also be calculated based on nautical charts and other information available, *apriori* to the mission. The global path is divided into way points.

6.1.2 The second level

At the second level, a local path planner plans a collision free path between the obstacles in the sonar's field of view, but obstacles that have been detected in the vicinity of the AUV should also be taken into consideration. The goal point is the next way point. The algorithm should be made so that a shortest path between the obstacles is found. It's important that the algorithm is fast.

6.1.3 The third level

At the third level there should be a pure reactive collision avoidance function that makes evasive manoeuvres when an object is found too close to the AUV. Then, there is no time to calculate a new path at the second level without risking that a collision arises. This can happen for example when an object is detected very late due to e.g. low level of the echo from the object.

6.1.4 The interaction between the levels

The second level and the third level are active between the way points that the first level has computed. A suggestion is that a new optimal path can be calculated when the collision avoidance function of the second level isn't able to find a free path out from clustered environments, or when the deviation from the original path is too large, or simply to keep the energy consumption down. The global path planner should be active at different time intervals, as often as the computer system on board the AUV permits it to. It should then consider all the obstacles the AUV has encountered.

6.2 The collision avoidance algorithm

In this project a collision avoidance function of the third level was made to get a feeling of how a collision avoidance algorithm should be worked out, given the behaviour of the vehicle's dynamics. This was done because most of the algorithms that have been studied in the literature have neglected or partly neglected the dynamics and have totally concentrated on the collision avoidance function. It has also been shown on AUV's that have been built in reality, that a simple collision avoidance function works better than the more complex ones [6].

The collision avoidance algorithm used in this thesis, makes evasive actions based on the data delivered from the sonar. If there is no risk of a collision then the collision avoidance algorithm heads for the next way point.

The algorithm was first developed in two dimensions, in the horizontal plane and was then expanded to three dimensions, involving all six degrees of freedom for the AUV.

6.2.1 Collision avoidance functions for two dimensions

Three different functions have been elaborated. Coarsely they can be explained as follows. The AUV is imagined to be surrounded with a safety zone. It means that if an obstacle is encountered within this zone, an evasive action should be made. The safety zone is chosen so that a complete turn, i.e. a course change by 180° can be made without risking that a collision occurs. The safety zone's radius is dependent on the horizontal sector of the sonar's field of view. With a narrower horizontal sector the radius have to increase. It also depends on the AUV's velocity. The evasive action is a new bearing (based on the location of obstacles) and in the third algorithm the velocity is also changed. The safety zone is set before the simulation. As the safety zone decreases the bolder the AUV becomes, i.e. the AUV gets closer to the obstacles before it does an evasive action. In the second and in the third algorithm the AUV has two safety zones. If the obstacles are within the outer safety zone, the AUV makes the same degree of evasive action as in the first algorithm and if it is encountered within the inner safety zone a stronger evasive action is made. In the third algorithm the velocity is changed based on in which part of the field of view the obstacle is located in. If there is no obstacle in the outer or the inner safety zone the AUV is heading for the next way point with maximum desired velocity. If an obstacle is encountered in the outer zone but not in the inner zone, the velocity is halved and if the obstacle is in the inner zone the velocity is halved once more.

6.2.2 Collision avoidance function for three dimensions

The collision avoidance strategy in three dimensions is based on the same principle as for two dimensions, that is the AUV is imagined to be surrounded by a safety zone, which in this case is a sphere. The algorithm calculates a new pitch angle and yaw angle based on where in the field of view the obstacle is encountered. If the obstacle covers up the horizontal field of view and the sonar can see above it or below it, the AUV tries to climb over it or dive under it.

7 Simulation environment

A simulation environment for studies of a collision avoidance function for AUV has been built up. The simulation environment will be explained in this chapter.
To be able to simulate several things are needed, such as:

- A simulation program. In this simulation environment, Matlab and Simulink are used.
- A model of the AUV. This could be a static or a dynamic model. If a dynamic model is used, then an autopilot to keep a given course and a given depth is also needed.
- Obstacles and a model of the sonar.
- An implementation of a collision avoidance algorithm that can be connected to the simulation program.

7.1 Vehicle dynamics and autopilot

The vehicle dynamics involves six degrees of freedom, which are x-, y- and z-direction, yaw (ψ), pitch (θ) and roll (ϕ) in a body-fixed coordinate system. This is illustrated in Fig. 7.1 and Fig. 7.2., where the angular velocity components p, q and r also are shown. u, v and w are the velocity components in the body-fixed x-direction, y-direction and z-direction.

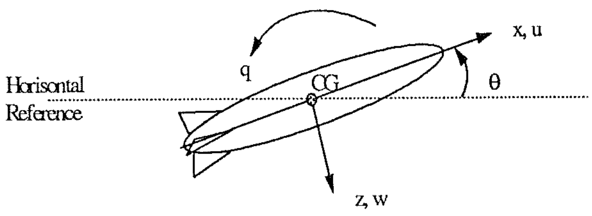


Fig. 7.1 The AUV from the side

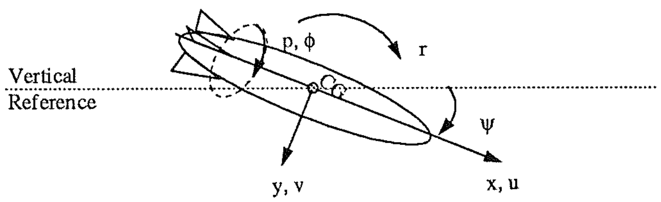


Fig. 7.2 The AUV from above

The model of the vehicle is a submarine model, where the non-linear differential equations of motion are given by the David Taylor Standard Submarine Equations of Motion [5]. Except from certain variables, the parameters of the model are dimensionless, which makes it easy to rescale the model. This was done resulting in a vehicle of 10 metres, which gives the same dynamic behaviour as the original submarine, but with a different time scale. A complete dynamic model of the AUV is then available.

Linearized models for course and depth are achieved by excluding terms that are non-linear in the equations of force and moment stated by David Taylor. The controllers in the autopilot are calculated with help of the linearized models, and are described in the following subsections. The linear models are valid for a constant forward velocity, u. Since the AUV is to travel with different velocities, this necessitates the use of gain scheduling with respect to u. The gain scheduler is implemented as a function in the program that uses linear interpolation to find gains for velocities that haven't been calculated apriori to the simulation. The control signals are the rudder angle δ_r, stern-plane angle δ_s, and bow-plane angle δ_b. No overshoot is wanted and the angles are limited to 30 degrees. The simulations in three dimensions are done with the complete non-linear AUV model. The validity of using a linear model for controller calculation was checked by a comparison of the closed loop performance with the non-linear model and the closed loop performance with the linear model, which didn't show any significant deviation.

The dynamic equations, linearization and depth controller calculation are described in more detail in [16].

7.1.1 Course Control

A linearization of David Taylor's equations for yaw rate r, and velocity v, along the body-fixed y-axis gives the model:

$$M_c \dot{\begin{bmatrix} v \\ r \end{bmatrix}} + D_c u \begin{bmatrix} v \\ r \end{bmatrix} = G_c u^2 \delta_r$$

A new state, uψ is introduced in the equation which gives

$$\dot{\begin{bmatrix} v \\ r \\ u\Psi \end{bmatrix}} = \begin{bmatrix} -M_c^{-1}D_c u & 0 \\ 0 \quad u & 0 \end{bmatrix} \begin{bmatrix} v \\ r \\ u\Psi \end{bmatrix} + \begin{bmatrix} M_c^{-1}G_c u^2 \\ 0 \end{bmatrix} \delta_r$$

By introducing $\delta_u = u\delta_r$, and for simplicity

$$A = \begin{bmatrix} -M_c^{-1}D_c & 0 \\ 0 \quad 1 & 0 \end{bmatrix}$$

$$B = \begin{bmatrix} M_c^{-1}G_c u \\ 0 \end{bmatrix}$$

$$x = \begin{bmatrix} v \\ r \\ u\Psi \end{bmatrix}$$

The equation can be written

$$\dot{x} = Aux + Bu\delta_u$$

The control law, $\delta_u = -Lx$, yields

$$\dot{x} = (Au - BuL)x$$

$$\dot{x} = (A - BL)ux$$

The closed loop poles are the eigenvalues of the matrix (A-BL)u. Varying u gives the same pole pattern, but poles with varying distance to the origin. Suppose that a suitable vector L has been found for a given constant velocity. The dependence $\delta_u = u\delta_r = -Lx$, which gives $\delta_r = -\frac{L}{u}x$, then states that the same L can be used for any velocity by scaling the original L with the chosen velocity. This yields the same behaviour of the closed loop, but with a different time scale.

7.1.2 Depth Control

A linearization of David Taylor's depth and pitch equations gives the model:

$$M_d \begin{bmatrix} \dot{w} \\ q \end{bmatrix} + D_d u \begin{bmatrix} w \\ q \end{bmatrix} + K_d \begin{bmatrix} \theta \\ z \end{bmatrix} = G_d u^2 \begin{bmatrix} \delta_s \\ \delta_b \end{bmatrix}$$

where δ_s is the stern-plane angle and δ_b is the bow-plane angle.
The given equation is then linearized for a constant velocity u, which gives :

$$\begin{bmatrix} \multicolumn{2}{c}{M_d} & 0 & 0 \\ & & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{w} \\ q \\ \theta \\ z \end{bmatrix} + \begin{bmatrix} D_d u \\ K_d \end{bmatrix} \begin{bmatrix} w \\ q \\ \theta \\ z \end{bmatrix} = G_d u^2 \begin{bmatrix} \delta_s \\ \delta_b \end{bmatrix}$$

For simplicity the model is written

$$\dot{x} = Ax + B\delta$$

Usually the autopilot in submarines is made so that the reference value is a desired depth. For the collision avoidance algorithm that has been tested, the wish was that the reference value should be a desired pitch in order to make the trajectory more smooth. The requirements of the control is then to follow a certain pitch, θ_r without any velocity w in the body-fixed z-direction. To achieve this a state-feedback gain, L, is computed from the states w, q and θ. The gain from θ_r to δ is called l_r. The controller can then be written:

$$\delta = l_r\theta_r - Lx$$

which gives

$$\dot{x} = (A - BL)x + Bl_r\theta_r$$

The output is given from

$$\begin{aligned} y_\theta &= [0 \quad 0 \quad 1 \quad 0]x = C_\theta x \\ y_w &= [1 \quad 0 \quad 0 \quad 0]x = C_w x \end{aligned}$$

A Laplace transform then yields

$$Y = C_i[sI - A + BL]^{-1}Bl_r\theta_r$$

where $i=\theta$, w.
In stationarity the aim is to achieve $\theta = \theta_r$ and w=0. This gives conditions for computing l_r:

$$\begin{aligned} C_\theta[-A + BL]^{-1}Bl_r &= 1 \\ C_w[-A + BL]^{-1}Bl_r &= 0 \end{aligned}$$

The matrix L was found with linear quadratic regulator design with help from Matlab. The matrix Q was supplied by Kockums Submarine System Ltd. and is given by

$$Q = \begin{bmatrix} 100 & 0 & 0 & 0 \\ 0 & 1000 & 0 & 0 \\ 0 & 0 & 1000 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

A suitable matrix R was found by trial and error and is given by

$$R = \begin{bmatrix} 200 & 0 \\ 0 & 200 \end{bmatrix}$$

7.1.3 Evaluation of the influence of the vehicle dynamics

A static model of the vehicles motion in two dimension is given by

$$\begin{cases} \dot{x} = u \cos \psi \\ \dot{y} = u \sin \psi \end{cases}$$

In this model the bearing can change instantly, which means that the AUV model doesn't set any limitations on the collision avoidance function. A sharp turn by $90°$, i.e. zero turning radius, is quite possible. With the complete dynamic model this behaviour isn't possible. The stand-off distance is the closest distance to an obstacle the AUV can have without risking a collision. The AUV has to keep a safe stand-off distance to the obstacles. When a turning is made to avoid an obstacle, there should be no risk of bumping into something that laid outside the sonar's field of view. Thus, the turning radius of the vehicle and the sonar's field of view are the limiting factors for the stand-off distance. As the vehicle's velocity changes, the turning radius changes, which means that the stand-off distance has to change.

Fig. 7.3.a illustrates the behaviour of the linearized course model with state feedback, obtained according to section 7.1.1. The AUV's velocity is 6 knots, i.e. approx. 3 m/s. In this simulation the AUV is initially located at the origin of the earth-fixed coordinate system, directed along the x-axis, i.e. the course angle (ψ) is zero. The desired course angle, $\psi_{ref} = 10°$, results in a time constant that is approximately five seconds and a delay of five metres before the course changes. When the velocity decreases the time constant increases, but the delay in metres doesn't change observably. This is due to the controller.

For changes in depth, the situation is somewhat different as the AUV can't make a loop. The reason for this is that the auxiliary equations stated by David Taylor are limited to a pitch angle, θ, in the interval $-90° < \theta < 90°$. However, in reality it is quite possible for the AUV to make a loop.

Fig. 7.3.b shows the closed loop behaviour of the linearized depth model with the velocity of 6 knots, with the feed-back gain computed from the matrices Q and R given in section 7.1.2. The goal is to follow the pitch $\theta_{ref} = 10°$. This means that the AUV should decrease its operating depth, see Fig. 7.1. Again the time constant increases as the velocity decrease. This means a longer time delay before the AUV changes the depth. Thus, when elaborating the collision avoidance algorithm, the things to be considered are: the sonar's field of view, the stand-off distance, the turning radius and the delay before the course and the depth changes.

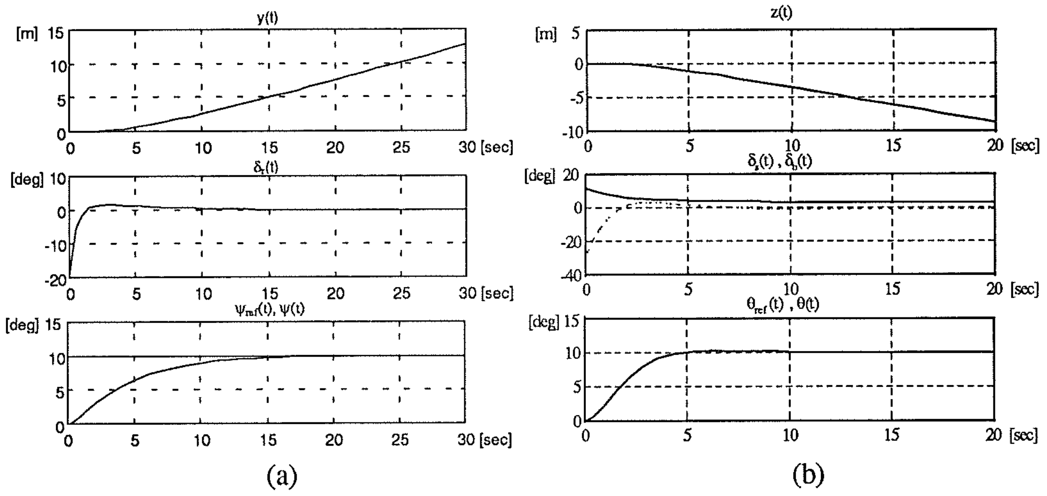


Fig. 7.3 The performance of state feed-back controllers obtained by (a) a course model and (b) a depth model of the AUV.

7.2 Assumptions on the sonar model and obstacles

To be able to simulate an implementation of a collision avoidance algorithm, a model of the vehicle and a model of the sonar are needed. The vehicle's model was discussed in section 7.1. The simulations were carried out with an ideal sonar model, i.e. the model didn't have any dynamics nor was there any processing of the data it would have produced. The intention was to get knowledge about what requirement the vehicle dynamics and the collision avoidance algorithm sets upon the sonar system. Once this is known, a sonar model can be elaborated and simulated.

The sonar could see all the obstacles that were within the field of view. Different values of the three parameters that defines the sonar's field of view, see Fig. 4.1., were tested to evaluate their influence on the performance of the collision avoidance function. The assumptions on the sonar model are very ideal as they overlook both how the raw data from the sonar look like and the processing of signals and images that has to be done before an obstacle can be identified. The effect of uncertainties like undetected obstacles due to threshold settings, non-returning sonar beams and other detection problems were however integrated in the sonar model.

The obstacles were given coordinates in the earth-fixed coordinate system for the central point and a radius for the extent. This radius was assumed to include uncertainties in the detection such as absence of echoes due to variations in the structure of the obstacle's surface. It may seem a little restricted that all the obstacles are represented as a circle/sphere, but a circle/sphere could be drawn around an obstacle of any geometry (a reef could for example be divided in many sections represented by a circle/sphere each). However, it simplifies the representation a great deal.

The obstacles position and extension are read from a file when the simulation starts. To decide if an obstacle is within the field of view of the sonar, the obstacle coordinates are first transformed to the body-fixed coordinate system. The algorithm then checks if any part of the obstacle is in the field of view. To transform the obstacle's coordinates x_{obs}, y_{obs} and z_{obs} to the body-fixed coordinates of the AUV, they are first translated by the

AUV's coordinates x_{auv}, y_{auv} and z_{auv}. As the AUV could have different yaw, pitch and roll angles, which are ψ, θ and ϕ respectively, the transformation also means a rotation of these angles. It's a standard, see e.g. [7], to first rotate ψ along the earth-fixed z-axis. This is achieved by the rotation matrix $R_{Z,\psi}$ which is defined by

$$R_{Z,\psi}=\begin{bmatrix}\cos\psi & \sin\psi & 0\\ -\sin\psi & \cos\psi & 0\\ 0 & 0 & 1\end{bmatrix}$$

The next step is to rotate the transformed matrix the angle θ around the y-axis. This is done by the rotation matrix $R_{Y,\theta}$, which is given by

$$R_{Y,\theta}=\begin{bmatrix}\cos\theta & 0 & -\sin\theta\\ 0 & 1 & 0\\ \sin\theta & 0 & \cos\theta\end{bmatrix}$$

The last step is to rotate with the matrix $R_{X,\phi}$, which means a rotation with the angle ϕ around the x-axis. The matrix $R_{X,\phi}$ is given by

$$R_{X,\phi}=\begin{bmatrix}1 & 0 & 0\\ 0 & \cos\phi & \sin\phi\\ 0 & -\sin\phi & \cos\phi\end{bmatrix}$$

The complete transformation of the obstacle's earth-fixed coordinates x_{obs}, y_{obs} and z_{obs} to the body-fixed coordinate system of the AUV, $x_{obs,AUV}$, $y_{obs,AUV}$ and $z_{obs,AUV}$ is then given by

$$\begin{bmatrix}x_{obs,AUV}\\ y_{obs,AUV}\\ z_{obs,AUV}\end{bmatrix}=R_{X,\phi}R_{Y,\phi}R_{Z,\psi}\begin{bmatrix}x_{obs}-x_{AUV}\\ y_{obs}-y_{AUV}\\ z_{obs}-z_{AUV}\end{bmatrix}$$

Based on these new coordinates, the sonar model used in the simulation decides which of the obstacles that lay in the sonar's field of view. The uncertainty is implemented such that objects can remain unseen under certain intervals of time. These intervals can be chosen by the user. Moving obstacles weren't simulated, but by expanding the data structure that defines the obstacles with a velocity and a direction, moving obstacles could easily be simulated. The problem with moving obstacles lays in the collision avoidance algorithm.

7.3 Computer implementation

The AUV's dynamics, the depth control, the course control, and the collision avoidance function are implemented in Simulink. The collision avoidance logic is programmed in C and is connected to Simulink through a mex-file. The collision avoidance logic is a discrete time system that calculates a new desired velocity, pitch angle and yaw angle. Each of the boxes in the figure below can easily be exchanged.

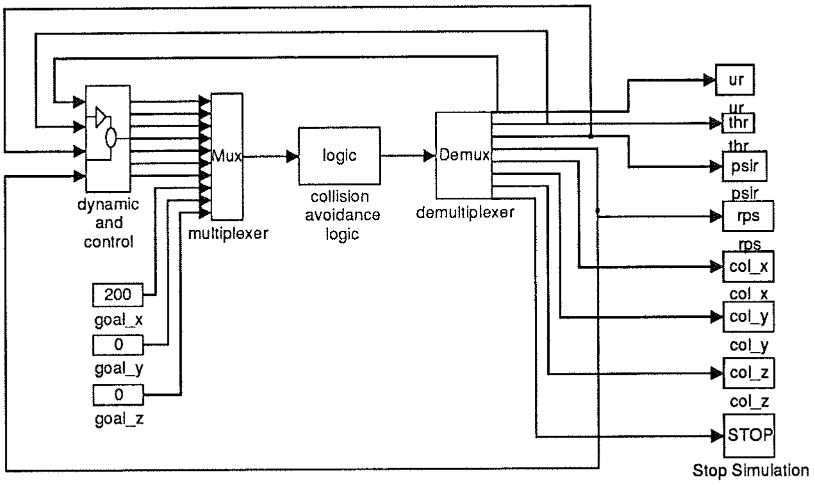


Fig. 7.4 Computer implementation

The collision avoidance logic contains the sonar model as well. The sonar model can easily be exchanged as long as the call in the C-code stays the same. The sonar's field of view is defined by the user through the Matlab variables range, alfa and beta. Goal_x, goal_y and goal_z defines the next way point, which is specified by the user. Ur, rps, psir, and thr are the desired values of velocity, number of revolutions of the propeller, course angle and pitch angle. Theses variables are calculated by the collision avoidance function, either to avoid a collision or to head for the next way point. The position of a collision, if a such has occurred, is given by col_x, col_y and col_z. The block stop simulation stops the simulation when the AUV is within a user-defined radius near the next way point. The block dynamic and control contains the dynamics of the vehicle and the autopilot with gain-scheduling. The parameters in the gain scheduler can be changed by the user.

8 Simulation examples

The simulations are done between two way points of the planned route. Different configurations of obstacles were tested to see how the collision avoidance algorithm performed. The simulations were first carried out in two dimensions. The simulation environment was then expanded to three dimensions, involving all six degrees of freedom for the AUV. The linear model of the AUV was used in the simulations in two dimensions, whereas the non-linear model was used for the simulations in three dimensions.

All the simulations that are illustrated here use a sonar with a range of 150 metres. However, the collision avoidance algorithms are made so that the evasive action isn't active as long as the distance to an obstacle is greater than 60 metres. Thus, for these simulations it is quite possible to use a sonar with a smaller range, but 150 metres was chosen because it is a range that is possible to find commercially. The distance of 60 metres was found suitable for the collision avoidance algorithm. A longer distance makes the algorithm more conservative. The lower limit of the distance is set by the AUV dynamics.

8.1 Simulation in two dimensions

The collision avoidance function steers out a new bearing every second. In the third algorithm where the velocity changes, a new desired velocity is also steered out. When the AUV travels with the maximum velocity, which is eight knots, this means that the AUV has moved approximately four metres since the last time the collision avoidance steered out new reference values. This has to be taken under consideration, as well as the delay before the bearing changes and the time constant needed for a new yaw reference, that were mentioned in section 7.1.3. If the AUV's velocity is high, less steering manoeuvres are done compared with if the velocity is low, as the collision avoidance function steers out new reference values every second. When the velocity is lowered, the AUV can keep the needed distance to the obstacles, due to the AUV moves shorter distances between the steered out reference values. Another consequence of this is that the AUV can set the course for the next way point earlier, when there is no danger of collision.

In the following figures the trajectory of the AUV is shown as the path of the centre of gravity of the AUV. Thus, at every point of time the AUV extend five metres in front of and five metres behind every point. In some of the simulations the dotted lines shows the width of the AUV. Initially the AUV is located in the point (0,0) in the earth-fixed coordinate system, and is directed along the x-axis.

To see how the three methods behave compared to each other, the energy expenditure and the length of the AUV's path is evaluated. The energy expenditure , e_c, is calculated as

$$e_c = \int \delta_r^2 d\delta_r$$

where δ_r is the rudder angle. Note that this energy measure only takes into account the effects of rudder movements. The effect of propulsion are not included.

There is no point in comparing the energy expenditure of the third algorithm with the first and the second algorithms because the AUV travels with different velocities in the third algorithm, whereas the velocity is constant for the two former.

It will be shown how the AUV reacts, when the horizontal sector of the sonar's field of view is changed and how the dynamics of the AUV affect the behaviour when the obstacles lay close to the AUV.

8.1.1 An obstacle that lay close to the AUV

The examples in this subsection are only given for the first algorithm as the two others behave in a similar way.

The obstacle is located in (25,25) and has a radius of 10 metres. The next way point is (60,60). The velocity of the AUV is 8 knots. Fig. 8.1.a shows the performance when the sonar's horizontal sector is $120°$. The obstacle is detected early, and the AUV can make a reactive manoeuvre to keep a safe distance to the obstacle. When the horizontal sector of the sonar is reduced to $60°$ the obstacle lays on the border of the sonar's field of view. Thus, the sonar doesn't see as much of the obstacle as in the former case, leading to that the AUV can't be so precautious and gets closer to the obstacle, see Fig. 8.1.b.

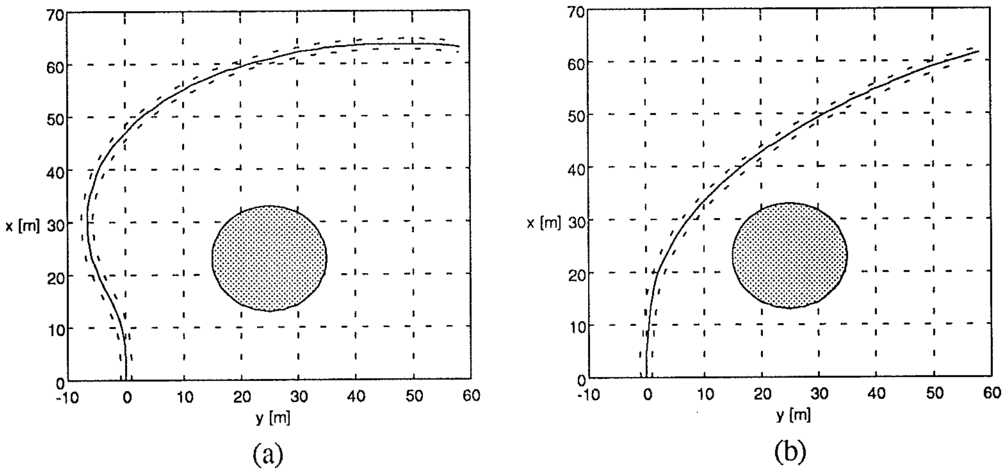


Fig. 8.1 The AUV's path for a sonar with a horizontal sector of (a) $120°$ and (b) $60°$. The AUV's velocity is 8 knots.

8.1.2 The influence of the controller

When the situation in Fig. 8.1.a is simulated with a controller that results in a time constant that is twice as big, the AUV's manoeuvrability decreases, see Fig. 8.2.a. The AUV isn't able to reach the next way point (60,60). This implies that the distance between the way points has to increase so that the AUV can reach the next way point without a large error of position. The deteriorated manoeuvrability is due to the larger time constant and not to the delay in bearing, as this doesn't change observably. However, the controller has less importance when the velocity is lowered, as the AUV moves shorter distances between subsequent steering operations. The advantage of a slower controller is that the energy expenditure decreases.

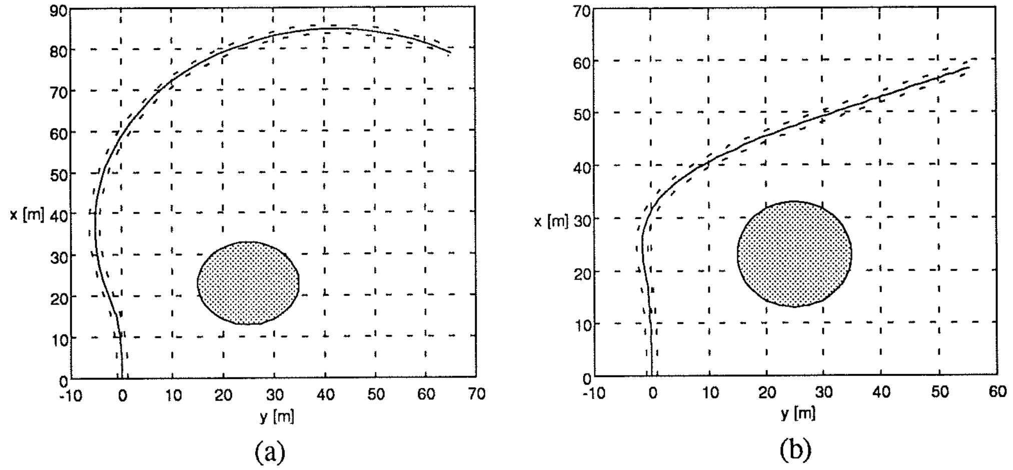


Fig. 8.2 (a) Simulation with a slower control
(b) Missed detection.

8.1.3 Missed detection

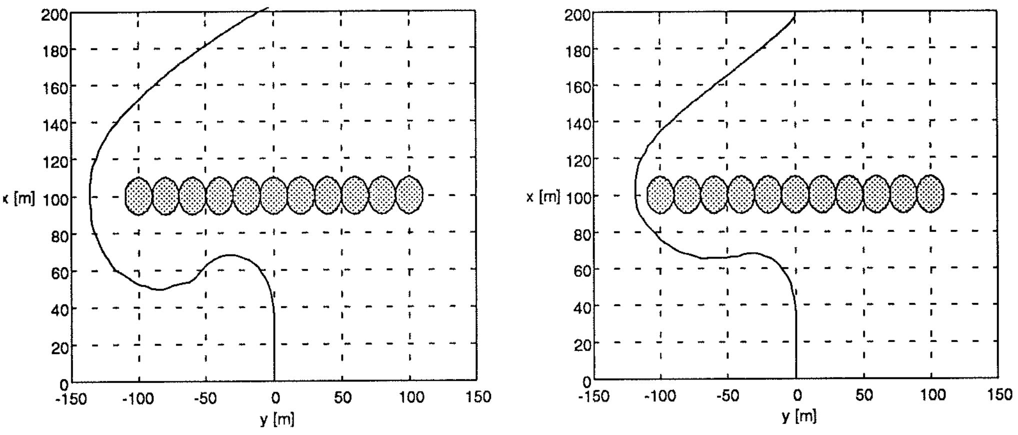
The obstacle detection is, as said before, affected with uncertainties depending e.g. on the transmitted beam's inclination angle on the obstacle's surface. This may lead to that the energy in the echo is too small to be detected or if it is detected, the signal level may be so low, that when the image is processed the obstacle isn't detected.

In this simulation the sonar is affected with uncertainty. The uncertainty is simulated by detecting the object only every second time it lays in the field of view of the sonar. This leads to that the AUV gets closer to the obstacle. Naturally, the situation gets worse if the obstacle is unseen for a longer period of time. Fig. 8.2.b shows the same situation as in Fig. 8.1.a, i.e. a sonar with a horizontal sector of 120°, but the simulation is done with the uncertainty.

8.1.4 The row

In this simulation the obstacles are lined up on a row in the earth-fixed coordinate system. There are eleven obstacles all having the same x-coordinate and a radius of ten metres. The goal-point is (200,0). The three algorithms are compared concerning the path length and energy expenditure. Different velocities of the AUV are simulated.

In Fig. 8.3.a the first algorithm is simulated. The AUV's velocity is 8 knots and the sonar's field of view is 120°. When the obstacles appear in the AUV's safety zone, the AUV turns away from the obstacles as long as there is a possibility of a collision. When this possibility disappears the AUV heads for the next way point. The resulting path length is 399 metres with an energy expenditure of 16.83. If the sonar's horizontal sector is halved, the AUV sees less of its surroundings, leading to that the AUV can't be so precautious as in the previous simulation. Fig. 8.3.b illustrates the situation. The path length decreases to 347 metres and the energy expenditure decreases to 15.9. This is because the sonar sees less of the environment, and the AUV gets closer to the obstacles before they are seen.



(a)

(b)

Fig. 8.3 The AUV's velocity is 8 knots and the sonar's horizontal sector is in (a) 120° and in (b) 60°.

When the AUV's velocity is decreased to 2 knots the path length decreases to 384 metres for a horizontal sector of 120° and to 336 metres for horizontal sector of 60°, to the expense of a higher energy expenditure, 33.76 and 18.55 respectively. This is due to that the collision avoidance function steers out new reference values four times as often, before the AUV has reached the corresponding distance, as in the simulations with a velocity of eight knots.

Several simulations have shown that for obstacles that lay closely to the AUV it's sufficient with a sonar with a horizontal sector of 60°. As all three algorithms are developed for a reactive collision avoidance function, the following simulations will all have a sonar with this horizontal sector. The second algorithm was simulated for velocities of 2, 4, 6 and 8 knots of the AUV. The minimum value of the path length is 334 metres and the maximum value is 343 metres for increasing velocities. With this configuration of obstacles there isn't a considerable difference of the energy expenditure compared to the first algorithm, if the velocity is high. The energy expenditure increases with decreasing velocity.

When the third algorithm is used, i.e. the velocity changes depending on the closeness of the obstacles, the mean value of the velocity is 5.8 knots instead of the desired velocity of 8 knots. The changes in velocity in the algorithm results in a somewhat shorter path length, 340 metres. Fig. 8.4.a shows the AUV's path for the third algorithm with a sonar's horizontal sector of 60°. If the second method is simulated with the resulting mean value of the velocity of the third algorithm, the path will be longer, 343 metres, compared to the third algorithm.

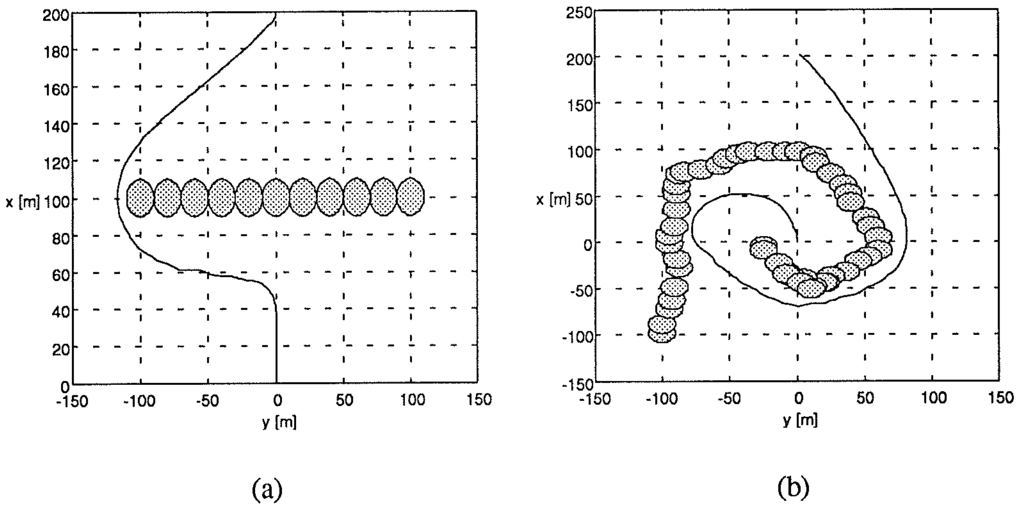


Fig. 8.4 (a) Simulation of the third algorithm with a desired velocity of 8 knots.
(b) Simulation of the first algorithm. The AUV's velocity is 8 knots.

8.1.5 The shell

In this simulation the obstacles configuration is shown in Fig. 8.4.b for the first algorithm with a horizontal sector of 60° of the sonar. The velocity is 8 knots and the next way point is (200,0). The path length is 590 metres and the energy expenditure is 26.66. If the second algorithm is simulated, with the same velocity of the AUV and the same field of view of the sonar, the path length decreases to 564 metres and the energy expenditure increases with 6 % comparing with the first algorithm. If the third algorithm is used with a desired velocity of eight knots the path length decreases to 557 metres. Due to the evasive actions needed, the AUV's velocity has a mean value of 5 knots. The AUV's path is shown in Fig. 8.5.a. If the second algorithm is simulated with the velocity of 5 knots the path length will be 560 metres.

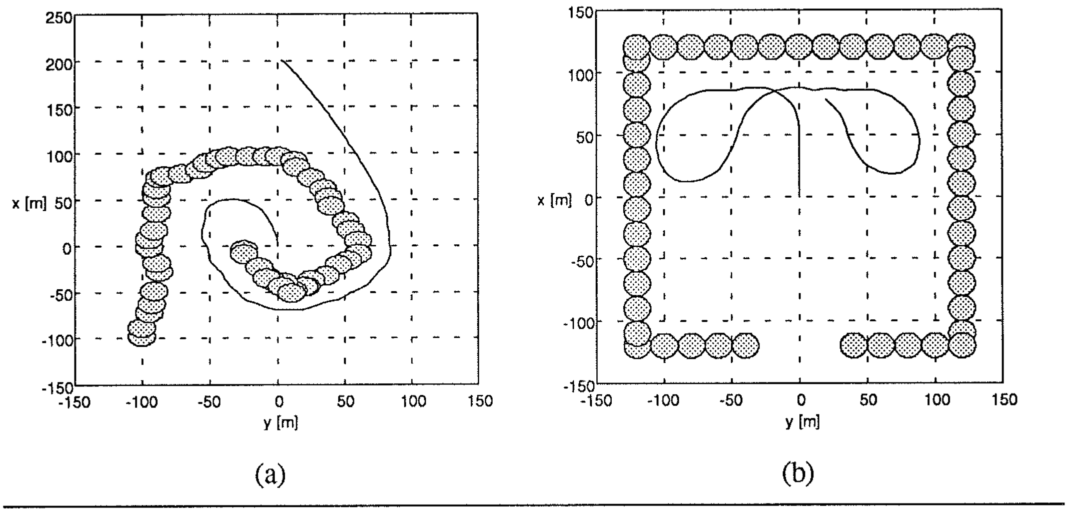


Fig. 8.5 (a) Simulation of the third algorithm with a desired velocity of 8 knots.
(b) No way out.

8.1.6 No way out

In this simulation the AUV travels with a velocity of 8 knots and the next way point is (300,0). The obstacles configuration is illustrated in Fig. 8.5.b. The AUV isn't able to find a way out from these obstacles, no matter which of the algorithms is used. This is a situation that may occur for the potential field method too, if there is no solution of getting trapped in a local minima. The reason for why the AUV can't find a way out is that the collision avoidance function makes evasive actions only when there are obstacles within the sonar's field of view, that may lead to a collision. When the AUV is located at the point (20,-100) there's no risk of colliding with an obstacle and the AUV heads for the next way point. Approaching this point the AUV sees a wall of obstacles and turns away. Then, the situation repeats and the AUV is trapped.

8.1.7 Comparison of the algorithms

If the three algorithms are compared, several simulations have given that the second algorithm yields a shorter path than the first algorithm. The third algorithm, which also changes the velocity of the AUV, gives the shortest path. The difference in the path length in the illustrated simulations doesn't seem important, but when the distance between the way points increases, the deviation between the path length will be more significant.

The reason for the resulting shorter path when the third algorithm is used, is that the velocity is lowered in a critical time when there is a risk of collision, and not because of the overall velocity is lower. This has been shown by simulations of the second algorithm, which is the same as the third but with a constant velocity. The price to be paid for a shorter path is that stronger evasive actions are needed and/or a change of velocity, which increases the energy expenditure. An energy calculation of a longer path versus stronger evasive actions with or without changes of the velocity, has to be done to evaluate which algorithm is best. This hasn't been done.

8.2 Simulation in three dimensions

When all six degrees of freedom for the AUV is used for avoiding a collision, there can be different ways of avoiding an obstacle. In this section it will be shown how the problem of not finding a way out, as was shown in Fig. 8.5.b, can be circumvented. Only two different simulations in three dimensions will be shown in this subsection, as it is a bit difficult to get a three dimensional feeling from the images on paper. The AUV is located in the origin of the earth-fixed coordinate system and is directed along the x-axis. In these simulations the sonar has a range of 150 metres, a horizontal sector of $60°$ and a vertical sector of $30°$.

8.2.1 The row

The obstacle configuration in this example is the same as in Fig. 8.3. The z-coordinate in the earth-fixed coordinate system is zero for all the obstacles. The obstacle radius is ten metres. The next way point is located in (300,0,0). Fig. 8.6 shows that by using all six degrees of freedom for the AUV, the AUV can find a shorter way by "climbing" over the obstacles.

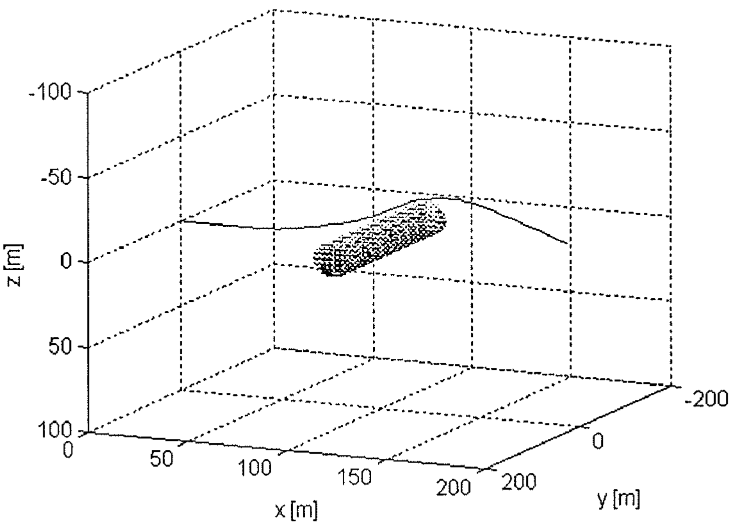


Fig. 8.6 The row in three dimensions. The AUV's velocity is 8 knots.

8.2.2 How a way out is found

In subsection 8.1.6 the simulation showed that the AUV was caught and couldn't find a way to the next way point. This could be solved by introducing a path planner on the second level, as was described in subsection 6.1.2. However, the problem can be circumvented in an easier way if all degrees of freedom for the AUV are feasible. This is shown in Fig. 8.7.

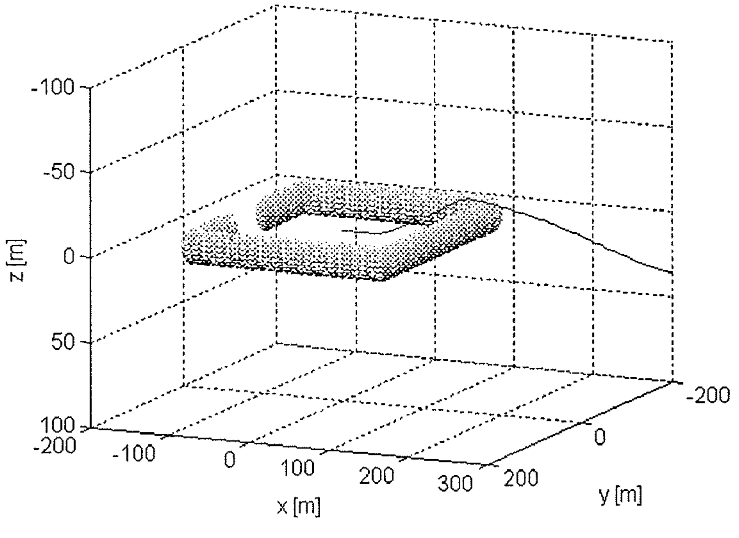


Fig. 8.7 The AUV's velocity is 8 knots. The AUV is able to find a way out.

8.3 Evaluation of the simulations

It has been shown that a simple collision avoidance function can perform well, given that the planned path is divided in way points, where the collision avoidance function is active. The simulations have shown that a sonar with a range of 150 metres, a horizontal sector of 60° degrees and a vertical sector of 30° gives a good performance. Sonars with these parameters can be found commercially.

The choice of a collision avoidance function that steers out new reference values every second originates in the study of sonar systems. To give an example, the sonar SE500 from Systems Engineering, can be used for collision avoidance. This sonar has a range of 100 metres and a 60° scanned horizontal sector size (two dimensions). It has a scan rate of two frames per second which depends on the range. This means that the image is updated every half second. It has to be kept in mind, that the images can change a great deal from one frame to the other depending on uncertainties discussed in chapter 4. After the image is updated from the sonar, it has to go through image processing. The positions and sizes of the obstacles has to be determined. This and the fact that the AUV is to travel with a velocity of 8 knots are the reasons why the collision avoidance algorithm is reactive and that new reference values are steered out every second. Thus, there is a very close relationship between the sonar and the collision avoidance function as can be expected.

If the AUV is to keep a specified depth position during the mission, i.e. evasive actions are only feasible in the horizontal plane, the collision avoidance function has to be elaborated so that it can handle the situation of 8.1.6. One advantage of only allowing evasive manoeuvres in one plane is that it is sufficient to do the object detection in a two dimensional sonar image instead of a three dimensional one. If all degrees of freedom of the AUV are used to make evasive actions, then it is necessary that there is a three dimensional image of the surrounding. If there is no restrictions on the AUV's motion, then it has been shown that a simple collision avoidance algorithm can be used for obstacle avoidance in three dimensions.

9 Concluding remarks

With the knowledge acquired from this thesis, the main problem seems to be extracting information from data delivered by the sonar. This is where the most effort has to be put in to get as good values as possible for the obstacles position and size. Accurate position values means that a good path planning can be implemented in the collision avoidance function to avoid unforeseen obstacles, which can minimise the energy consumption of the AUV, when an evasive action has to be done. This fact is very important.

Another problem is considered to be moving obstacles. This issue is important to deal with, especially when the AUV is heading for the surface of the sea, and there is a danger of colliding with ships. More research has to be done in that area.

Yet another problem is the navigation error, which in the specification of this project is 50 metres per hour travel. The navigation accuracy sets limits upon the global path planner, which after a while only will work as a reference for the direction, leading to that the second level and the third level described in section 6.1 will be more active.

To have a good collision avoidance function at the second level, it is very important that the obstacles are detected as soon as possible, so that a new path can be calculated. This means that the range of the sonar should be large, so that the collision avoidance function can avoid obstacles in an intelligent way. But as the range becomes larger the resolution decreases. This means that smaller objects can remain undetected which increases the possibility of a collision. It also means that the objects that are detected have a greater uncertainty in calculated position and extension. Then, the AUV has to keep a larger distance to the detected obstacles. This in turn affects the energy consumption, which is limited on an AUV. Thus, a high resolution is wanted to make a good collision avoidance function, but if the resolution is high, the field of view has to be scanned more often and with a smaller beam width, which results in a longer time to update the frames. It also means a higher data rate to the image processing. As the resolution decreases, the possibility of making a good collision avoidance function also decreases, which could mean that instead of planning a free path it only generates an evasive impulse when an obstacle is detected.

9.1 Future work

If this project would have continued the focus would have been on developing a good sonar model to be used in the simulation. A method of processing an image from the sonar has to be elaborated so that it can be used in the simulation. The next step should be to find a model for the navigation system. If the design is made as suggested in chapter 3, then the obstacle logger and the environment update should be implemented. When all the models are found and all the implementations are done, then there is a solid foundation for testing more advanced collision avoidance functions that contains some degree of path planning too. Once this is done, a thorough validation of different collision avoidance function can take place.

10 References

1. Ayela G., Bjerrum A., Bruun S., Pascoal A., Pereira F-L., Petzelt C., Pignon J-P. "Development of a Self-Organizing Underwater Vehicle - SOUV". Marine Sciences and Technologies, Second MAST days and Euromar market —Volume II.
2. Bergem O. "A multibeam sonar based positioning system for an AUV". Proceedings of the 1993 symposium in Unmanned Untethered Submersible Technology, Portsmouth, NH, 1993.
3. Bjerrum A., Ishoy A., Henriksen L. " Sonar-based Navigation of an Unmanned Untethered Underwater Vehicle". Hydroakustik 1995, Sweden,12-13 September, 1995.
4. Conte G, Zanoli S.M. " Obstacles avoiding in unknown environment: An issue for underwater autonomous vehicles". Proceedings of the 1993 symposium in Unmanned Untethered Submersible Technology, Portsmouth, NH, 1993.
5. David W. Taylor Naval Ship Research and Development Centre. "DTNSRDC revised standard submarine equations of motion".
6. Ferguson J and Pope A. " 'Theseus': Multipurpose Canadian AUV". Sea Technology April 1995.
7. Fossen T.I. "Guidance of Ocean Vehicles". John Wiley Sons 1994.
8. Gonzalez R.C., Woods R.E. "Digital Image Processing". Addison-Wesley Publishing Company.
9. Henriksen L. "Real-Time Underwater Object detection Based on an Electrically Scanned High-Resolution Sonar". Proceedings of the 1994 Symposium in Autonomous Underwater Vehicle Technology, Cambridge, 1994.
10. Hyland J.C. " Optimal obstacle avoidance path planning for autonomous underwater vehicles ". Proceedings of the 1989 symposium in Unmanned Untethered Submersible Technology, Portsmouth, NH, 1989.
11. Hyland J.C. " A comparison of two obstacle avoidance path planners for autonomous underwater vehicles". Proceedings of the 1990 symposium in Autonomous Underwater Vehicle Technology, Washington, DC ,1990.
12. Jircitano A. "Gravity based navigation of AUV's". Proceedings of the 1990 symposium in Autonomous Underwater Vehicle Technology, Washington, DC ,1990.
13. Khatib O. "Commande dynamique dans l'espace opérational des robots manipulateurs en présence d'obstacles". Ph.D dissertation, École Nationale Supérieur de l'Aéronatique et de l'Espace (ENSAE), France, 1980.
14. Kim Jin-Oh and Khosla P.K "Real-time Obstacle Avoidance Using Harmonic Potential Functions".IEEE Transactions on Robotics and Automation, vol 8, NO 3 June 1992.
15. Loggins C.D " Ahead-look sonars: Design comparisons and preformance considerations". Sea Technology, March 1995.
16. Nilsson H. "Cross-coupling Effects in Submarine Control". Master Thesis project, Lund Institute of Technology, Sweden, 1995.
17. Porto W.V., Fogel D.B., "Neural Network Techniques for Navigation of AUV's". Proceedings of the 1990 Symposium in Autonomous Underwater Vehicle Technology, Washington, DC ,1990.
18. Shipmate+, Marine Electronics " GPS Satellite Navigation".
19. Urick R.J. "Principles of Underwater Sound", 3rd edition, McGraw-Hill Book Company.

20. Warren C.W. "A Technique for Autonomous Underwater Vehicle Route Planning". Proceedings of the 1990 Symposium in Autonomous Underwater Vehicle Technology, Washington, DC ,1990.

21. Williams G.N., Lagace G.E. and Woodfin A.J. "A collision avoidance controller for autonomous underwater vehicles". Proceedings of the 1990 Symposium in Autonomous Underwater Vehicle Technology, Washington, DC ,1990.