

Institutionen för systemteknik

Department of Electrical Engineering

Examensarbete

Controlling an autonomous underwater vehicle through tunnels with a behavior-based control strategy

Examensarbete utfört i Reglerteknik
vid Tekniska högskolan vid Linköpings universitet
av

Olle Axelsson

LiTH-ISY-EX--11/4519--SE

Linköping 2011



Linköpings universitet
TEKNISKA HÖGSKOLAN

Controlling an autonomous underwater vehicle through tunnels with a behavior-based control strategy

Examensarbete utfört i Reglerteknik
vid Tekniska högskolan i Linköping
av

Olle Axelsson

LiTH-ISY-EX--11/4519--SE

Handledare: **Martin Skoglund**
isy, Linköpings Universitet

Sören Ohlsson
Saab Underwater Systems

Examinator: **Fredrik Gustafsson**
isy, Linköpings Universitet

Linköping, 20 October, 2011

Abstract

The objective of the master's thesis work is to investigate how an autonomous underwater vehicle (AUV) should act in an underwater tunnel environment. The thesis proposes sensors, control strategies, mission statement, among others, required for tunnel assignments.

A behavior-based control (BBC) strategy has been developed to control the AUV. The BBC is used in the middle level of the vehicle control, i.e. the reactive control system which describes how the AUV navigates through a tunnel, while other events are considered. The control strategy has also been separated into two parts, and these are: controlling the AUV's heading and controlling the AUV to a desired distance from the tunnel wall.

To be able to evaluate the performance of the system, a graphical user interface (GUI) has been developed. The GUI enables the operator to change control settings during simulations. Two proposed control strategies are presented with simulated results.

Sammanfattning

Syftet med examensarbetet är att undersöka hur en autonom undervattensfarkost (AUV) bör agera i en undervattentunnel miljö. Avhandlingen föreslår sensorer, reglerstrategier, uppdragsbeskrivning med mera som krävs för tunneluppdrag.

En beteendebaserad (behavior-based) reglerstrategi har utvecklats för att styra AUV:n. Reglerstrategin används i mellersta nivån i farkostens reglering, det vill säga den reaktiva regleringen som beskriver hur farkosten ska styra genom en tunnel samtidigt som andra händelser beaktas. Reglerstrategin har även delats upp i två delar: reglering av AUV:ns kurs och reglering av AUV:n till ett önskat avstånd från tunnelns vägg.

För att kunna verifiera funktionaliteten av systemet så har även ett grafiskt användargränssnitt utvecklats. Gränssnittet möjliggör att man kan ändra reglerparametrar under en simulering. Två föreslagna reglerstrategier presenteras med tillhörande resultat.

Acknowledgments

I would like to thank Saab Underwater Systems (SUS) for the opportunity of doing my thesis there. Several people at SUS have been involved in the thesis and are worth mentioning. I especially want to thank Henrik Welin and Nicklas Johansson for a lot of interesting discussions and valuable advice. Beyond those I want to thank Sören Ohlsson (supervisor) and Roger Johansson (first contact with SUS). Without them, the thesis would not be achievable. I would also like to thank the remaining people at SUS for a friendly and good meeting. Finally, I want to thank my supervisor Martin Skoglund and my examiner Fredrik Gustafsson for the feedback on the report.

Contents

1	Introduction	3
1.1	Background	4
1.2	Problem Formulation	4
1.3	Objective	5
1.4	Related Research	6
1.5	Outline	7
2	Background Theory	9
2.1	Operating an Underwater Vehicle	9
2.2	Behavior-Based Control	10
2.3	Existing Structure	11
	2.3.1 Components	12
	2.3.2 Mission Planner	12
2.4	Utility Function	13
2.5	Euler Angles	15
2.6	Sensors	16
3	Implementations	17
3.1	Sensors	18
3.2	New Components	18
	3.2.1 Way-Point Control	19
	3.2.2 Tunnel Inspection	21
	3.2.3 Visualizer	21
	3.2.4 Graphical User Interface	22
	3.2.5 Mission Planner	23
3.3	Controllers	24
	3.3.1 Algorithm 1 : Lateral Control	24
	3.3.2 Algorithm 2 : Heading Control	25
4	Results	29
4.1	Tuning Parameters	29
4.2	Scenarios	30
	4.2.1 Wall Tracking	30
	4.2.2 Tunnel Tracking	32
	4.2.3 Complex Mission	35

5 Summary	39
5.1 Conclusions	39
5.2 Further Developments	40
Bibliography	43
A Wall Tracking	45
B Tunnel Tracking	48
C Complex Mission	50

Abbreviations

AI: Artificial Intelligence

AUV: Autonomous Underwater Vehicle

BBC: Behavior-Based Control

GUI: Graphical User Interface

IMU: Inertial Measurement Unit

MMI: Man-Machine Interface

ROV: Remotely Operated Vehicle

SUS: Saab Underwater Systems

UAV: Unmanned Aerial Vehicle

UUV: Unmanned Undersea Vehicle

Chapter 1

Introduction

The following chapter introduces general information of this master's thesis work. It introduces the main problem to be solved, its background and the reason why a solution is needed.

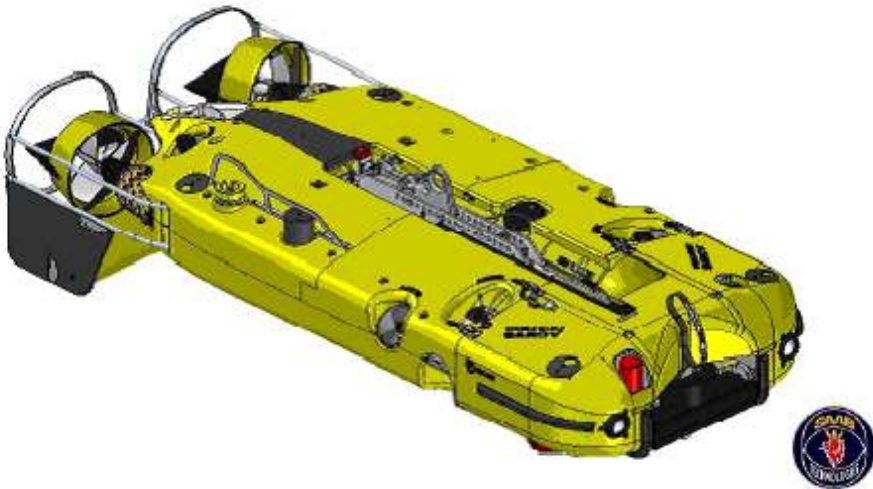


Figure 1.1. *The basis of the master's thesis work is the underwater vehicle Double Eagle SAROV. The vehicle can act both in tethered and untethered mode. Technical data: dimensions: 2950 x 1240 x 750 mm, weight in air: 710 kg, weight in water: slightly positive, max depth: 300 m, forward speed: 6 knots, lateral speed: 0.7 knots, vertical speed: 0.4 knots, forward thrust: 2500 N.*

1.1 Background

The thesis was performed at Saab Underwater Systems (SUS) in Motala, Sweden. SUS develops underwater vehicles that offer solutions for mine hunting, maritime security, underwater surveillance and unmanned missions. The most well-known Remotely Operated Vehicle (ROV) families are: the Double Eagle Systems and the Sea Owl Systems. [11]

Figure 1.2 and 1.3 illustrates two examples of the above mentioned ROV families.

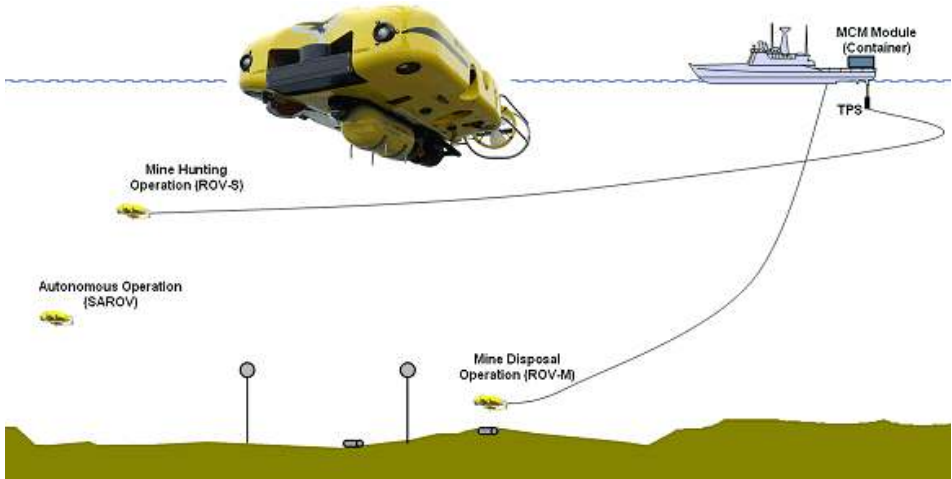


Figure 1.2. "The Double Eagle Concept uses a modular approach to provide flexible systems for MCM applications (Mine Hunting, Mine Disposal). It is aimed at providing a hydro dynamically stable, highly reliable system with exceptional performance and low life cycle cost." [13]

1.2 Problem Formulation

The problem to solve is how to navigate an AUV through tunnel environments.

Because of complex underwater environments, autonomous underwater vehicles (AUVs) have become necessary. One of the reasons is the limitation of the human capacity of handling all inputs from the vehicle surrounding environment. Another reason is the growing demands from the industry. Today an ROV may not be sufficient to fulfill all requirements. The reasons are many, for example: long missions, strong currents, small/dark/complex environments, among others. Together with an already tough start there is also a tether ¹ which should be long enough, taut all the time (to prevent tangles) etc. The operator must also be constantly active due to all kinds of events, and the error margins may be small.

¹A tether is a cable that provides power and control signals to the vehicle.

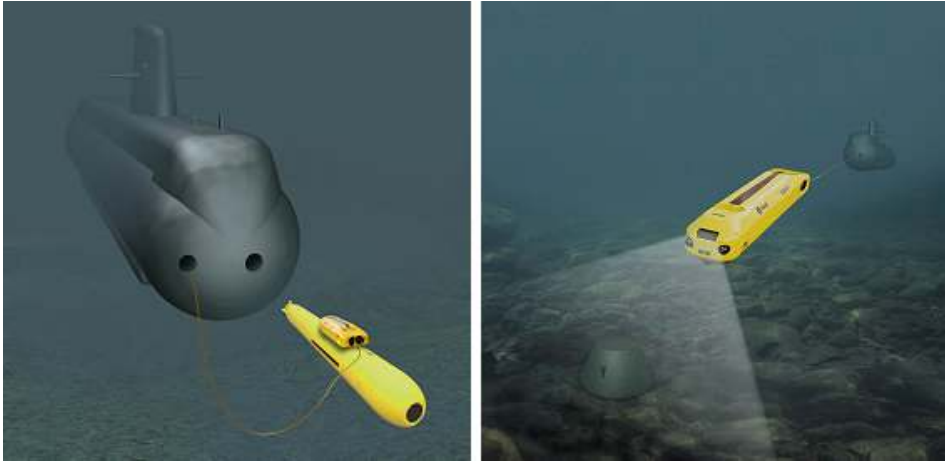


Figure 1.3. *"The newest member in the Sea Owl family is the Sea Owl SUBROV system. This new system is intended for use from a submarine torpedo tube. The SUBROV has capability for Inspection, UW-works, and Mine Counter Measures, as a platform for Communication/Surveillance and as an active Docking Tool for an AUV." [16] The left figure illustrates the ROV docked at a torpedo tube and the right figure a mine searching ROV in front of the submarine.*

The above mentioned cannot be handled by an ordinary operator. Instead of controlling the vehicle manually a solution is to transfer responsibility from the operator to the AUV. This can be managed using a higher level of mission planning, which must be configured offline before a mission. The beneficial properties for the introduction of an AUV are many, but several other problems arise, such as: battery time, mission planning and safety, which have to be solved in a good manner.

1.3 Objective

The objective of the master's thesis work is to investigate how an autonomous underwater vehicle (AUV) should act in an underwater tunnel environment. To limit the thesis, three main parts are set in focus, and these are: propose and evaluate the sensors needed for a tunnel inspection, propose and evaluate algorithms for the middle level of the vehicle control and finally propose and evaluate additions to the top level of the vehicle control. The middle level contains of the reactive control system which describes how the AUV navigates through a tunnel, while other events are considered and the top level control is for the management of the mission statement.

The final result will be proposed algorithms that should work in a real-time system and therefore have an acceptable calculation time. The algorithms will be implemented in today's software and tested in a simulator. The algorithms are developed and evaluated using Microsoft Visual Studio (C++) and Matlab.

1.4 Related Research

There are a lot of researchers on autonomous vehicles, for example: P. Kachroo and S. Wadoo have written the book, *Autonomous Underwater Vehicles: Modeling, Control Design and Simulation* [6] which offers examples of underwater vehicle construction, exploring kinematic fundamentals, problem formulation, and controllability, among other key topics. The book *Automatic vehicle guidance* written by A. Broggi, M. Bertozzi, A. Fascioli and G. Conte introduces a prototype car called ARGO and its setup and performance [1]. The book *UAV Cooperative Decision and Control* written by Steven J. Rasmussen and Tal Shima describes unmanned aerial vehicles (UAVs) [9]. The above shows the breadth of what might be in the area of autonomous vehicles. There are also other books that investigate control strategies, not just in an autonomous vehicle. For example: the professor Thor I. Fossens proposes different methods for controlling ocean vehicles in his book, *Guidance and Control of Ocean Vehicles* [10].

To control an autonomous vehicle, it does not matter if it is a flying, diving or road vehicle, several common problems arise. For example: map-building, control strategy and required sensors.

One method for building the environment is by an occupancy grid. The occupancy grid maps the environment as an array of cells. The occupancy grids were introduced by H. Moravec and A. Elfes in the 1980s. They have written a lot of articles, for example: *High resolution maps from wide angle sonar* [3] and *Using occupancy grids for mobile robot perception and navigation* [2].

An example of a book describing the principles, design, and practice of intelligent behavior-based autonomous robotic systems is written by Ronald C. Arkin [8]. An interesting PhD thesis which summarizes different behavior-based control strategies is written by Michael R. Benjamin [5].

A more related PhD thesis about an autonomous underwater vehicle (AUV) is written by Marc Carreras Pérez [4]. The thesis introduces different control strategies and experimental results from the autonomous underwater vehicle URIS in a water tank. The robot URIS weighs 30 kg and is equipped with four thrusters.

No articles or books about tunnel inspection are found. But there are several companies developing tunnel inspection AUVs and remote operated vehicles (ROVs). The Oceanographic Systems Lab (OSL) has an autonomous underwater vehicle, built in 2004, that is designed for tunnel assignments. The vehicle purpose was to inspect (find its leaks) a 45 mile section of the Delaware Aqueduct [15]. The company, Hibbard Inshore has used ROVs in underwater work and inspection, applications in industries such as power plants, hydro electric dams, water conveyance tunnels, ocean outfalls, bridges, recovery and near shore cable lay [12].

There are also a lot of interesting projects around the world that investigates autonomous vehicles, for example: *US military DARPA Grand Challenge* [22] and *Google driverless car* [21].

1.5 Outline

A behavior-based control model has been developed to handle tunnel assignments. Figure 1.4 illustrates the model. The model consists of four behaviors², and these are: stabilize heading, follow track, avoid obstacle and tunnel inspection, which together represents the reactive part in the control system. The reactive part describes how the vehicle navigates through the tunnel, while other events are considered. Chapter 2 describes the behavior-based control model further.

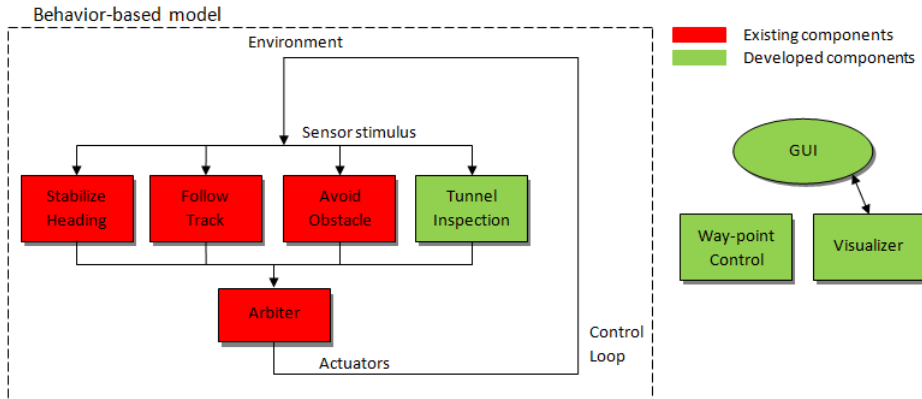


Figure 1.4. The above figure shows an overview of the behavior-based control model and three other components used in the control. There are already a couple of existing components, developed by SUS, and new components developed especially for tunnel assignments. The way-point controller and the visualizer are used for the tracking of the way-points and for the visualization of the vehicle.

A simple example showing the use of the behavior-based control is illustrated in Figure 1.5. The vehicle has two behaviors which should give the AUV a new best heading direction.

The proposed algorithms control the yaw (heading) angle and the lateral velocity. The pitch and the roll angle are set to zero and are controlled by an already implemented controller. The main purpose is to control the AUV's heading and to control the AUV to a desired distance from the tunnel wall. In order to control the vehicle to a desired distance from the tunnel wall, a hyperplane has been added to the vertical plane of the AUV. To be able to control the heading, a hyperplane has been added to the horizontal plane.

²"A behavior encapsulates the perception, planning and task execution capabilities necessary to achieve one specific aspect of robot control, and receives only that data specifically required for that task." [7]

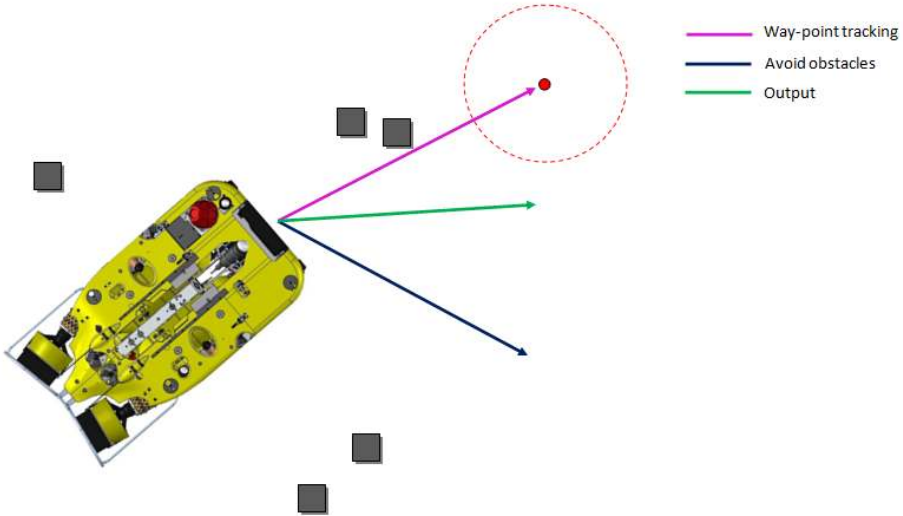


Figure 1.5. The above figure illustrates the performance of the behavior-based control model. In this scenario, the vehicle should try to reach a way-point while avoiding obstacles. To manage this there are two behaviors, one for way-point tracking and another for avoiding obstacles. The behavior-based control model calculates a "optimal" output (heading angle), depending on the behaviors.

The rest of the thesis is organized into four chapters: *Background Theory, Implementations, Results* and *Summary*. Chapter 2 provides the reader with general information of the existing system, which is needed for the understanding in next coming chapters. In Chapter 3 the implementations are described. Chapter 4 presents the results of the implemented components and Chapter 5 summarizes the master's thesis work and propose further work in the field of AUVs.

Chapter 2

Background Theory

The following chapter introduces general theory necessary for the next coming chapters. It introduces the investigated AUV, a control strategy called behavior-based control and other related theory used in the master's thesis work. Readers that are familiar with the described theory may continue to read Chapter 3.

2.1 Operating an Underwater Vehicle

In this thesis an AUV called Double Eagle SAROV is investigated. Figure 1.1 illustrates the AUV. The vehicle can operate in tethered and untethered mode and is mainly developed to handle missions in open water. Today the vehicle can act autonomous according to a stated mission plan. The mission plan has limited functionality; it can follow a track, avoid obstacles and act at different ways depending on the surrounding environment.

Several sensors are required to operate an underwater vehicle. A typical underwater vehicle is equipped with leak sensors, temperature sensors, cameras, sound velocity probe, sonars, inertial measurement unit (IMU), doppler-log, among others. It is possible together with computer software to control the vehicle by the thrusters. To protect above electrical equipments, all are set in a hull that can handle pressure differences.

The operator communicates with the vehicle through a tether. In untethered mode a mission plan is required before going into a mission, and the communication is handled by WLAN at surface or by radio modem or acoustic underwater modem. The communication links enables the autonomous underwater vehicle to get new commands and missions.

2.2 Behavior-Based Control

The ability to handle unpredictable events summarizes the behavior-based control model (BBC). It can for example be a person who is driving a car when an animal appears in front, and then the person must act fast and steer away from the upcoming collision. The ability to handle changing environments and unpredictable events is necessary in an autonomous vehicle.

The behavior-based control model is used because a complete model of the world is unreasonable. In comparison to the standard model where a model of the entire environment, or an approximation, is necessary. The disadvantage of the standard structure is the ability to react in unpredictable events where no model exists. [8]

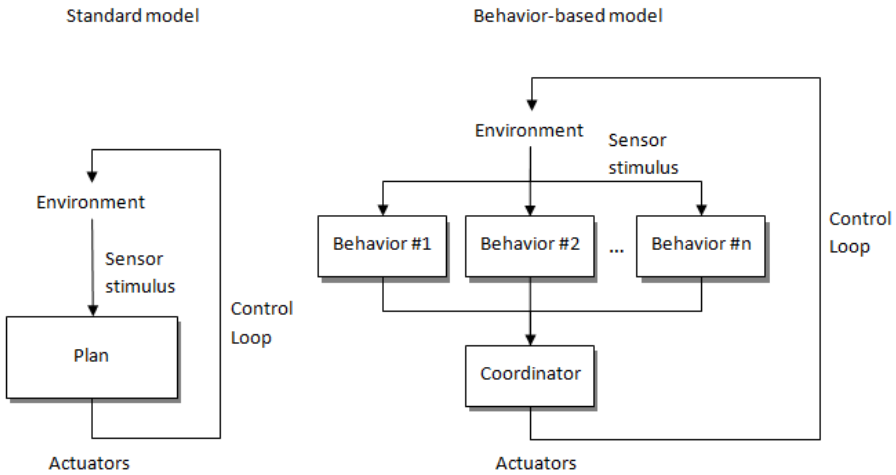


Figure 2.1. The following figure illustrates the standard model and the behavior-based model. The standard model consists of three modes: sensor, plan and act. The standard model must have a complete or approximate model of the world. The BBC structure is built on different behaviors and does not need an entire model of the world. The behaviors are weighted in the coordinator before receiving the actuators. It is easy to add new behaviors to the BBC model but it can be hard to weight the different behaviors in the coordinator.

The behavior-based model is composed of specialized task-achieving modules, or behaviors, that operate independently. A behavior receives only information for its specifically task and tends to respond directly to immediate stimuli. A simple figure of the BBC model and the standard model is shown in Figure 2.1. To be able to choose between the different outputs, from the behaviors, there are a coordinator. The coordinator can be a simple switch or have a more advanced structure, including artificial intelligence (AI).

2.3 Existing Structure

An overview of the already implemented autonomous parts of the vehicle system is shown in Figure 2.2. The system is developed by SUS and built in a BBC structure. The system consists of five components and serves as a platform when developing new components in the thesis. The components are: avoid obstacle, stabilize heading, follow track, track control and arbiter (coordinator), where the first three are called behaviors. The components have been developed in C++.

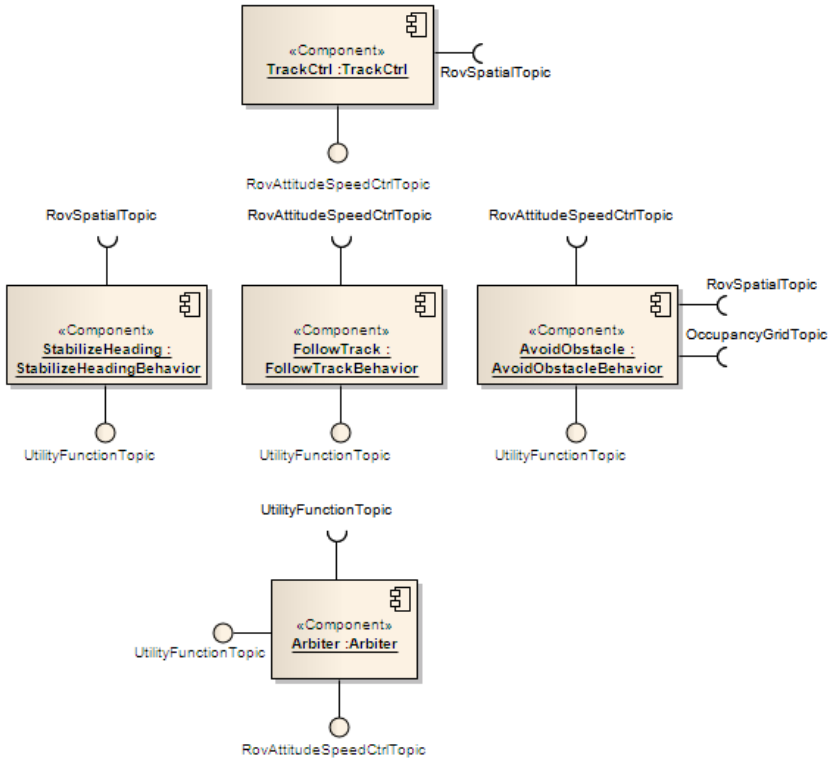


Figure 2.2. A snapshot of the already implemented autonomous parts (BBC components) in the AUV. It consists of five components, and these are: avoid obstacle, stabilize heading, follow track, track control and arbiter. The first three are called behaviors and these translate sensors outputs and mission commands from the mission planner (sent via the track control component), to a desired heading angle and velocities. Finally, the signals from the three behaviors are weighted in the arbiter (coordinator), and the outputs are final control signals used in the low level controller.

The outputs from the behaviors are built on utility functions, which are described in 2.4. The final control signals are calculated in the arbiter and are based on the utility functions sent from the different behaviors.

2.3.1 Components

The following subsection introduces each component in the current structure. Figure 2.2 illustrates the already implemented components by SUS.

Track Control: The track control component acts as a link between the operator and the AUV. The component receives way-points³ from the mission planner and translates it to a desired heading angle and velocities which are sent to the behaviors.

Follow Track: The follow track component converts the desired heading and velocities, sent from the track control, to appropriate control signals.

Stabilize Heading: The following component stabilizes the vehicle forward motion and protects the AUV to turn around during a mission.

Avoid Obstacle: In order to handle unpredictable events the avoid obstacle behavior acts as a safety device for the AUV. When an obstacle appears the behavior want to change heading and decrease the velocity of the AUV.

Arbiter: The final component acts as a coordinator. The arbiter collects data from the behaviors and summarizes it to control signals. To be able to choose which behavior is most important in the summation there is a priority level for each behavior. The priority level is set between zero and one where a higher value indicates in a more important behavior. The final result consists of a desired heading angle and velocities which are sent to a low level controller.

2.3.2 Mission Planner

The *mission planner* is a man-machine interface (MMI) developed by SUS. The MMI is built in Java in Netbeans and it enables the operator to configurate missions for the AUV. An example of a mission can be to investigate a harbor.

The MMI consists of several actions, and these are: transit, loiter, surface and optical survey. By combining these, the operator can configure different missions. Figure 2.3 shows an example of a mission. The purpose of this mission is to reach a way-point and then circulate around it.

³A way-point is a desired position in the "world". The way-point is expressed in global coordinates in a latitude, longitude and altitude or depth position.

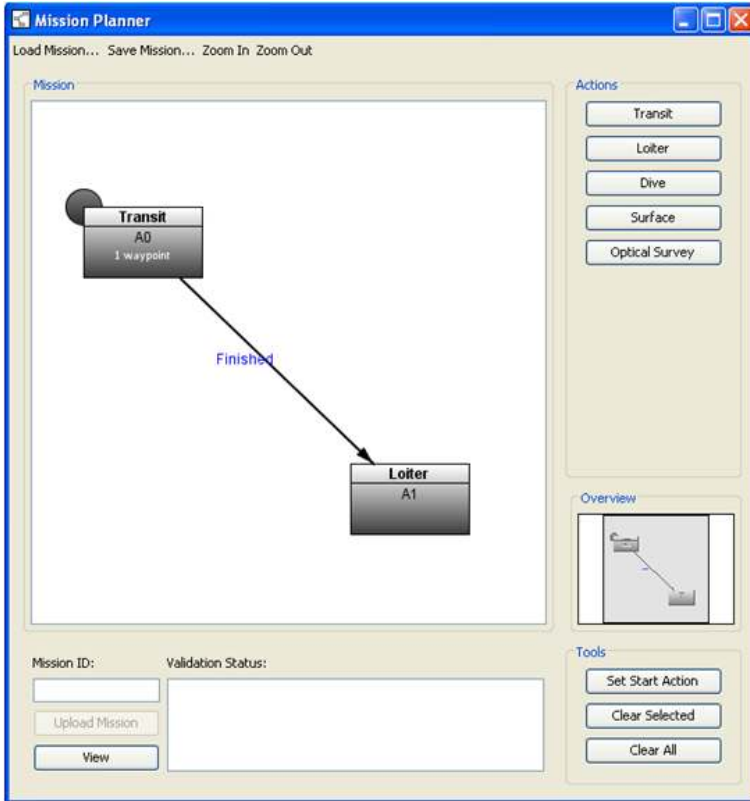


Figure 2.3. The following figure shows an example of a mission in the mission planner. The mission starts with a transit action and ends with a loiter action. The transit action consists of one way-point to be reached by the AUV. When reached the way-point, next action begins where the AUV circulates slowly around the way-point.

2.4 Utility Function

A *utility function* is a mathematical function describing some sort of "probability" of an object. The utility function is the underlying mathematical function when designing the algorithms presented in Chapter 3. It is based on a linear function defined by a rectangular peak and is expressed mathematically in (2.1). Figure 2.4 illustrates an example.

$$y(c, d, h, x) = \begin{cases} h - d(x - c) & \text{if } x > c \\ h + d(x - c) & \text{if } x < c \end{cases} \quad (2.1)$$

Input parameters: c - center, d - spread (derivative), h - activation level, x - control range interval

Output parameter: y - "probability"

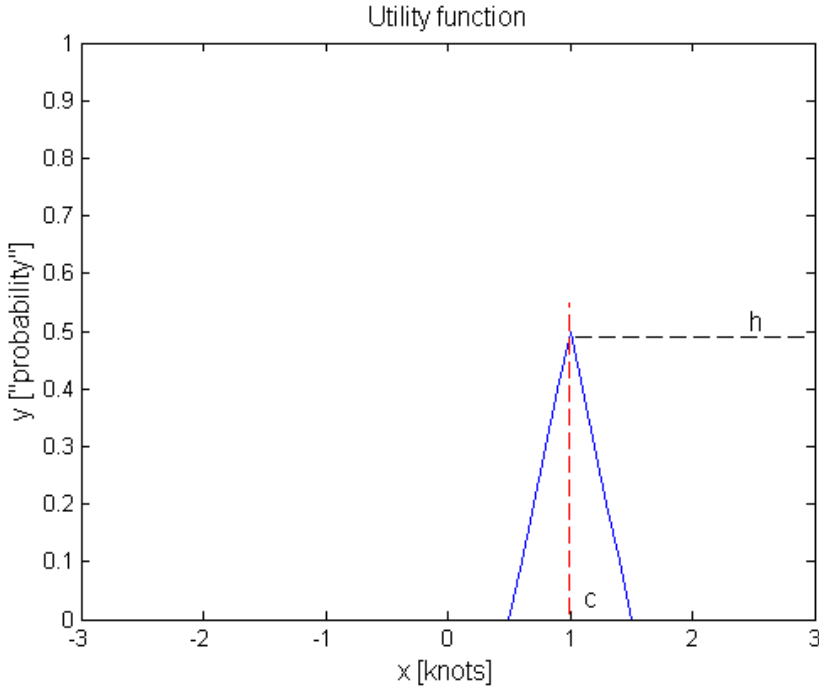


Figure 2.4. The following figure shows an example of a utility function which serves the basis for building the algorithms presented in Chapter 3. The utility function has a desired activation level (h), spread (derivative) and a position on the x-axis (c). In this case the utility function is used in the speed control where the control range interval is between -3 and 3 knots. The spread is set to approximate 16 percent which implies in a width between 0.5 to 1.5 knots for the utility function.

The activation level is the height of the function and can be seen as a "probability". The spread is given in percent between 0 and 100 and indicates how wide the utility function is. The control range interval is set between minimum and maximum control range.

Example 2.1: Utility Function

If there is an obstacle in the direction of 45 degrees, at range of 1 m, and a spread of 10% is used. The center of the utility function is then at 45 degrees and the height is proportional to the distance and in this case also negative for avoidance. The heading control range interval is between 0 and 360 degrees and therefore the utility function's width is between 9 and 81 degrees (center at 45 degrees).

2.5 Euler Angles

The *Euler angles* [17] are used for the implementation of a local coordinate system for the vehicle. It is based on three rotations; pitch (ϕ), roll (θ) and yaw (ψ). Each rotation can be expressed mathematically and together in a transformation matrix. The final rotation matrix is shown in (2.2).

$$Q = \begin{bmatrix} \cos \theta \cos \psi & -\cos \phi \sin \psi + \sin \phi \sin \theta \cos \psi & \sin \phi \sin \psi + \cos \phi \sin \theta \cos \psi \\ \cos \theta \sin \psi & \cos \phi \cos \psi + \sin \phi \sin \theta \sin \psi & -\sin \phi \cos \psi + \cos \phi \sin \theta \sin \psi \\ -\sin \theta & \sin \phi \cos \theta & \cos \phi \cos \theta \end{bmatrix} \quad (2.2)$$

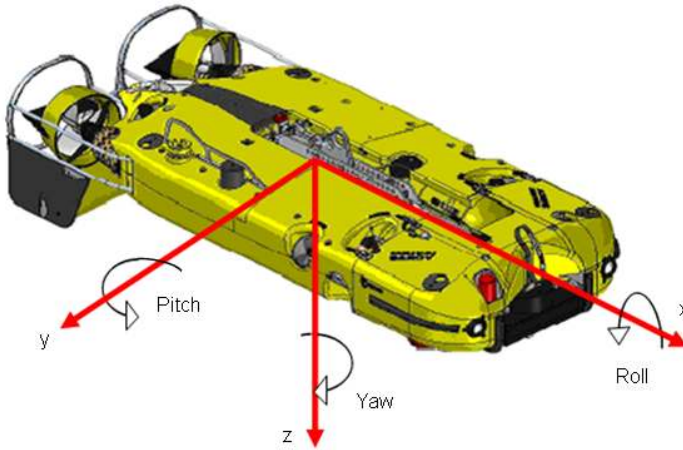


Figure 2.5. The following figure illustrates the AUV with its three Euler rotations; pitch, roll and yaw.

The system has 6 degrees of freedom, the motions in x, y and z directions respectively, and roll, pitch and yaw for the rotations about x, y and z axes respectively. The estimation of these angles are not part of the thesis, but are assumed to be known and well estimated. Figure 2.5 illustrates the AUV with its three Euler rotations.

The proposed control strategies control the yaw (heading) angle and the lateral velocity. The pitch and the roll angle are set to zero and are controlled by an already implemented controller.

2.6 Sensors

There are a lot of sensors in the area of underwater vehicles. One frequently used sensor is the single-beam sonar. The sonar is based on an echo signal, where a time difference is used for calculating the range and the bearing to an object/obstacle. It is possible together with the navigation system and the Euler angles to calculate the bearing and the range relative the vehicle to an obstacle. The limitation of these sonars is the sweep time. A regular sonar has a maximum sweep speed which depends on a predefined maximum echo range. It is a trade off between the sweep speed and the echo range. An introduction of sonars can be read on wikipedia [19]. Figure 2.6 illustrates the principle of a sonar.

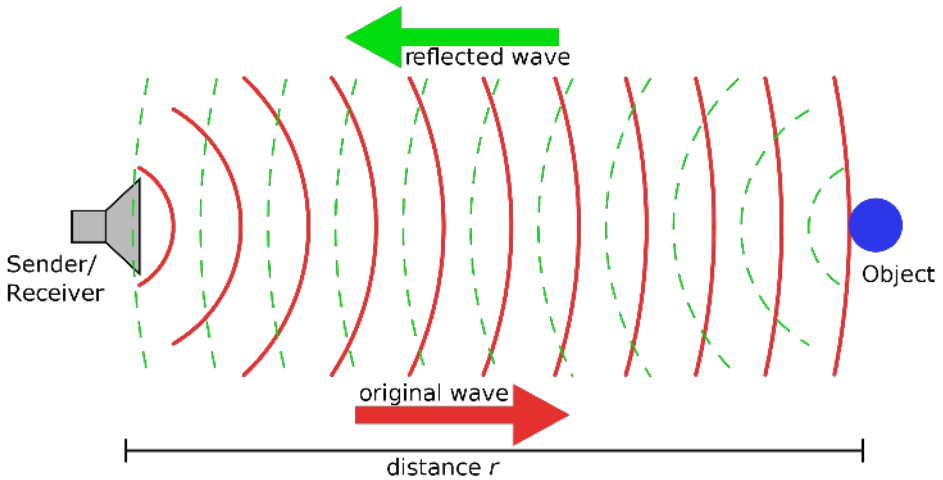


Figure 2.6. The following figure illustrates an active sonar. The sonar uses a sound transmitter and a receiver to detect obstacles. [19]

Alternative there are multi-beam sonars. A multi-beam sonar consists of several single-beam sonars, which can map more than one location on the seabed with a single ping and with a higher resolution than conventional echo sounders. The disadvantage is a more complex picture which need more signal processing and sensor fusion. The multi-beam sonar is often used together with cameras to get a better view of the environment.

Chapter 3

Implementations

The following chapter presents developed and implemented algorithms with components. The algorithms are based on the theory from Chapter 2. Figure 3.1 illustrates the new components with an associated signal flow.

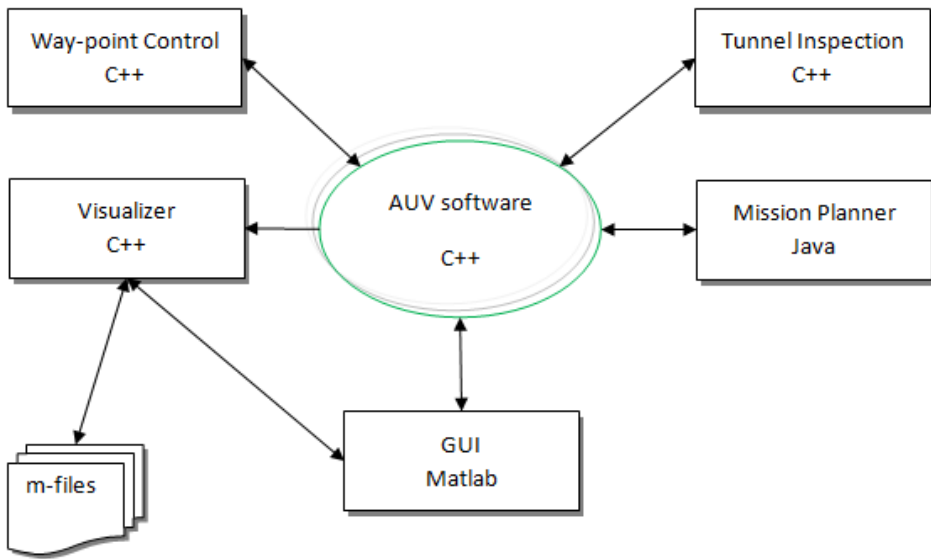


Figure 3.1. The following figure shows an overview of the signal flow between the implemented components. Three new components have been added to the existing system and a graphical user interface (GUI) for analyzing and debugging. The center of the system is the AUV software. The three new components are: the tunnel inspection, the visualizer and the way-point control. The visualizer is used for collecting data between the behaviors, the tunnel inspection is used with the way-point control for the navigation in a tunnel environment. There is also a GUI which is connected to the visualizer, the GUI enables the operator to change control settings during simulations.

3.1 Sensors

During the thesis two range sonars have been added to the simulator environment. The first sonar is sweeping in the horizontal plane and the other in the vertical plane.



Figure 3.2. The following figure illustrates the simulator environment where two sonars have been added to the AUV. The first sonar is sweeping in the horizontal plane and the other in the vertical plane. The sonars are based on echo signals and the outputs are a range and a bearing to an obstacle. The white boxes illustrate obstacles, hit by the sonars. As can be seen in the figure the AUV is in a tunnel environment that also has been developed during the thesis.

Figure 3.2 illustrates the simulator with the two sonars. The sonars have a sweep speed of 36 degrees/s and a hit range of 30 m. The vertical sonar is sweeping between 0 and 360 degrees and the horizontal sonar only in front of the vehicle (0 to 180 degrees).

3.2 New Components

Three new components have been developed during the thesis and these are: the way-point control, the tunnel inspection and the visualizer. In addition to the above, a graphical user interface has been developed.

3.2.1 Way-Point Control

The *way-point control* calculates a desired heading angle based on a way-point. The component is developed in Microsoft Visual Studio (C++). The operator expresses the way-point in the mission planner with a desired forward velocity. The way-point is described in global coordinates in a latitude, longitude and altitude or depth position. Figure 3.3 illustrates the AUV with a desired heading calculated by the way-point control component.

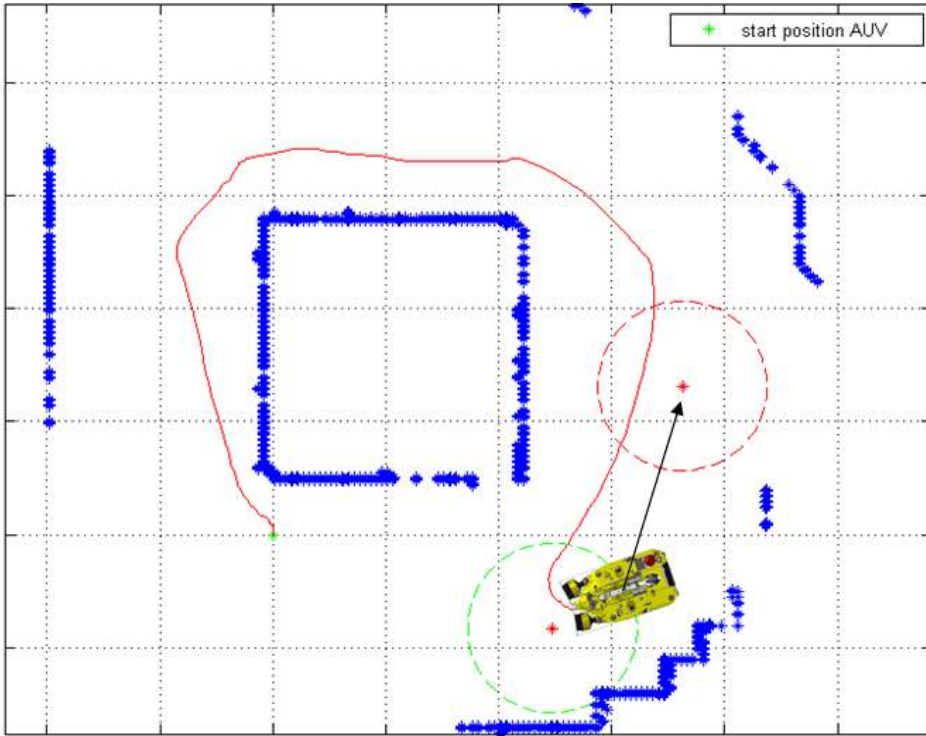


Figure 3.3. The following figure shows the AUV with the way-point control. The AUV has managed to reach a way-point and is on the way to the next. The arrow illustrates the desired heading angle calculated by the way-point control component.

The definition of reaching a way-point is set by the operator. The tuning parameter is a radius, defining a circle around the way-point. The way-point is reached when the AUV is in the circle. A simple sketch of how the component interacts with the rest of the system is shown in Figure 3.4. As can be seen in the figure the way-point control is connected to the follow track component where the desired heading and forward velocity are converted to appropriate control signals.

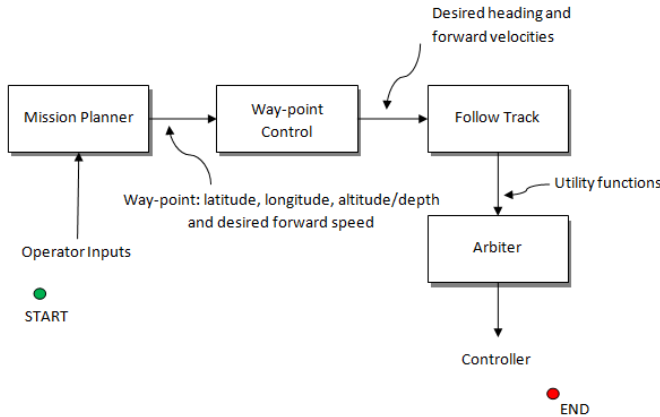


Figure 3.4. A simple figure showing the steps from the operator inputs to the actual control output. The operator defines the way-point or way-points in the mission planner. These are sent to the way-point control where a desired heading is calculated. The next step is to convert the heading to an appropriate control signal which is used in the arbiter and finally in the low level controller.

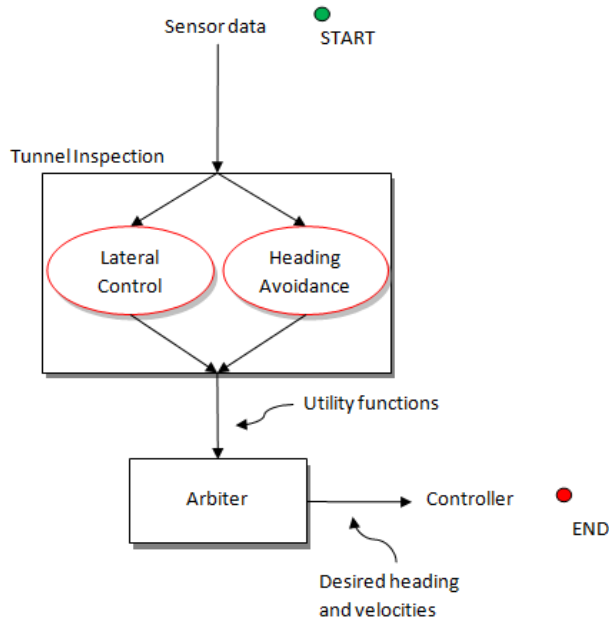


Figure 3.5. The following figure illustrates the tunnel inspection component. The tunnel inspection is built on two algorithms, one for heading avoidance and another for controlling the AUV to a desired distance from the tunnel wall.

3.2.2 Tunnel Inspection

The *tunnel inspection* component is especially developed to handle different missions in small environments. It consists of two algorithms that can be selected by the operator in the graphical user interface, and these are: controlling the AUV to a desired distance from the tunnel wall and controlling the heading angle of the vehicle.

The tunnel inspection should be used together with the way-point control and the follow track component during missions. Figure 3.5 shows a simple sketch of the component.

3.2.3 Visualizer

The *visualizer* is developed in C++ and it enables the operator to get part of the output signals in the different components. The purpose of this is mainly for debugging and monitoring during simulations. Figure 3.6 illustrates the component input and output signals.

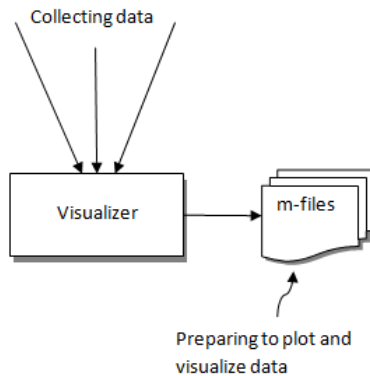


Figure 3.6. The following figure presents the input and output signals in the visualizer component. The component collects data to be analyzed in a couple of m-files. The m-files prepare to plot for example the utility functions and the way-points during a simulation/mission.

The signals are collected and stored in an automation Matlab session by the Matlab Engine [14]. The visualizer has several m-files⁴ connected to it. The m-files collect desired data and prepare to plot and analyze the motion of the AUV. The visualizer serves as a platform between the C++ language and the more mathematical language, Matlab. It enables the user to send data to Matlab for example debugging, analyzing and monitoring.

⁴The name of a Matlab function file begins with an alphabetic character and has a filename extension of .m.

3.2.4 Graphical User Interface

The *graphical user interface* (GUI) enables the operator to get information from the AUV. It is a tool for debugging and monitoring. It is developed and built in Matlab. The GUI is connected to the visualizer and controls the component. It enables the user to see and plot signals stored by the visualizer and there are also several predefined plot functions, which illustrates for example the utility functions, the way-points etc. From the beginning, the GUI was not planned to exist. But it was a necessary tool in the development of the algorithms and especially for the tuning of the control parameters. An overview of the GUI is shown in Figure 3.7.

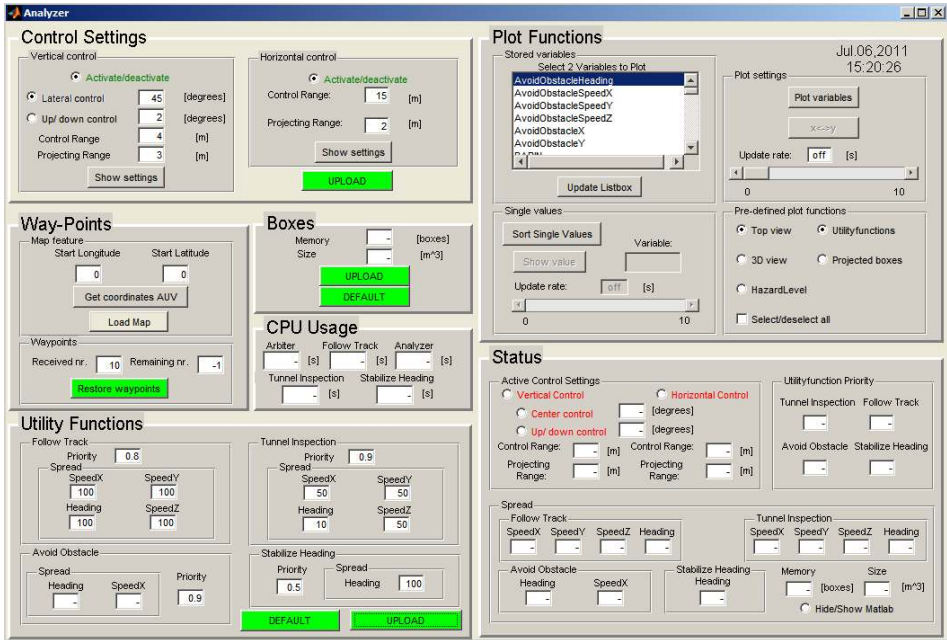


Figure 3.7. The following figure shows an overview of the graphical user interface. The GUI consists of seven areas, and these are: control settings, way-points, boxes, CPU usage, utility functions, plot functions and status. The area way-points enables the operator to see remaining and received number of way-points. It has also a map feature which enables the operator to upload a map and choose desired way-points. The plot functions area collects all variables stored by the visualizer component and it enables the operator to analyze and plot these signals. The status area indicates all defined settings in the AUV. The utility functions area enables the operator to change priority and spread in the different behaviors. The box area enables the operator to set how many and the size of the hit boxes. The control settings area enables the operator to change control strategies and settings, such as projecting range and control range. Finally, the CPU usage area shows the computation time for each behavior which has been very useful when integrating Matlab functions in C++.

The GUI has a map feature where the operator can upload a map (nautical chart) of an environment. The map should be two dimensional and in *.fig* format. Before choosing way-points, the start position of the AUV is needed. After reading this position from the AUV or manually write it, it is forward to use the map feature.

Except from analyzing data, the GUI enables the operator to change control settings and control strategies during simulations. It is for example possible to change priority and spread in the different behaviors.

3.2.5 Mission Planner

The extended version of the original MMI presented in Chapter 2 is illustrated in Figure 3.8. It has a developed tunnel inspection action where the user can specify desired way-points to be followed. The tunnel inspection action is integrated in the original software and works together with the other actions. Before sending the way-points to the AUV it is packed into a format called XML ⁵, and when received by the AUV software it is unpacked and used in the way-point control component.

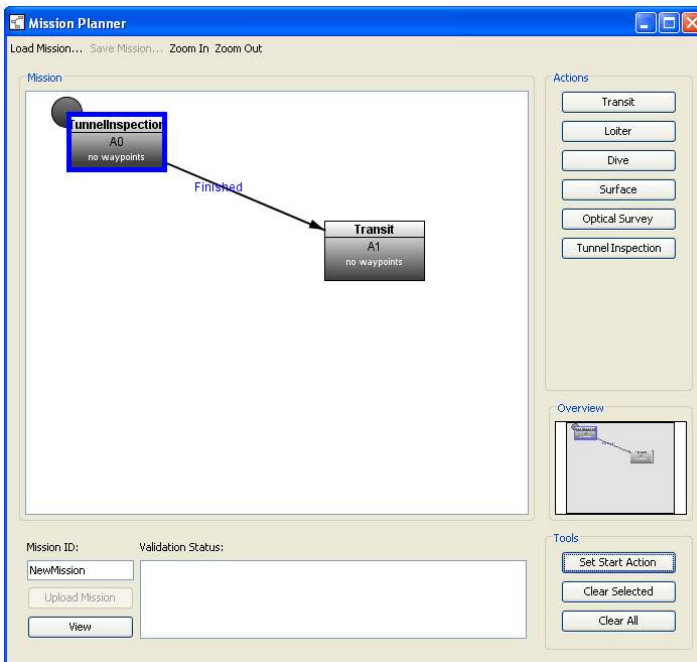


Figure 3.8. The following picture gives an overview of the extended MMI. A new action called tunnel inspection has been added where the operator can upload different way-points to be followed during missions.

⁵Extensible Markup Language (XML) is a set of rules for encoding documents in machine-readable form [20].

3.3 Controllers

The tunnel inspection algorithm is separated into two control strategies, and these are: controlling the AUV to a desired distance from the tunnel wall and controlling the heading angle of the vehicle. The algorithms are based on the utility functions and the Euler angles described in Chapter 2. The outputs from the algorithms are desired velocities and a heading angle. Figure 3.9 shows an overview of the dependencies between the algorithms and the rest of the system.

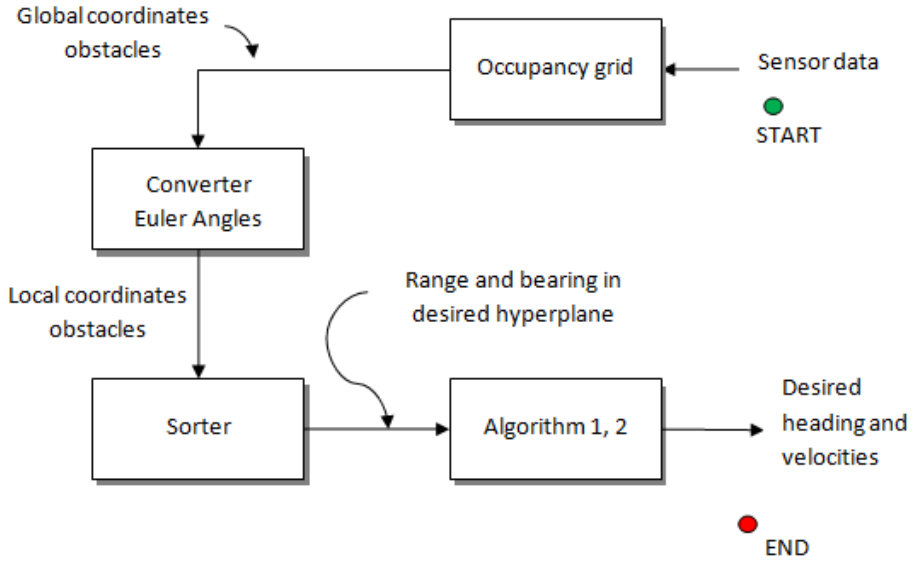


Figure 3.9. The following figure illustrates the signals from the sonars to the low level controller. Sensor data are sent to the occupancy grid for building an environment based on the hit points. The next step is to convert the global coordinates to local and before receiving the main algorithms, the obstacles are collected in different control zones by the sorter. Algorithm 1 and 2 calculates a desired heading angle and velocities before receiving the arbiter and finally the low level controller.

3.3.1 Algorithm 1 : Lateral Control

In order to control the vehicle to a desired distance from the tunnel wall, a hyperplane has been added to the vertical plane of the AUV. The vertical plane is defined by a circle around the vehicle. The tuning parameters are a circle radius and a projecting range. The projecting range works like a looking forward decision range and the circle radius as a range when to control the vehicle. After projecting the obstacles, all obstacles have a range and a bearing in the vertical plane. When building the utility function to control the vehicle, only the obstacle closest to the vehicle is used. The activation level of the utility function is proportional to the

range to the obstacle. An obstacle close to the AUV has a higher hazard level than an obstacle far away. Figure 3.10 illustrates the different control zones in the vertical plane.

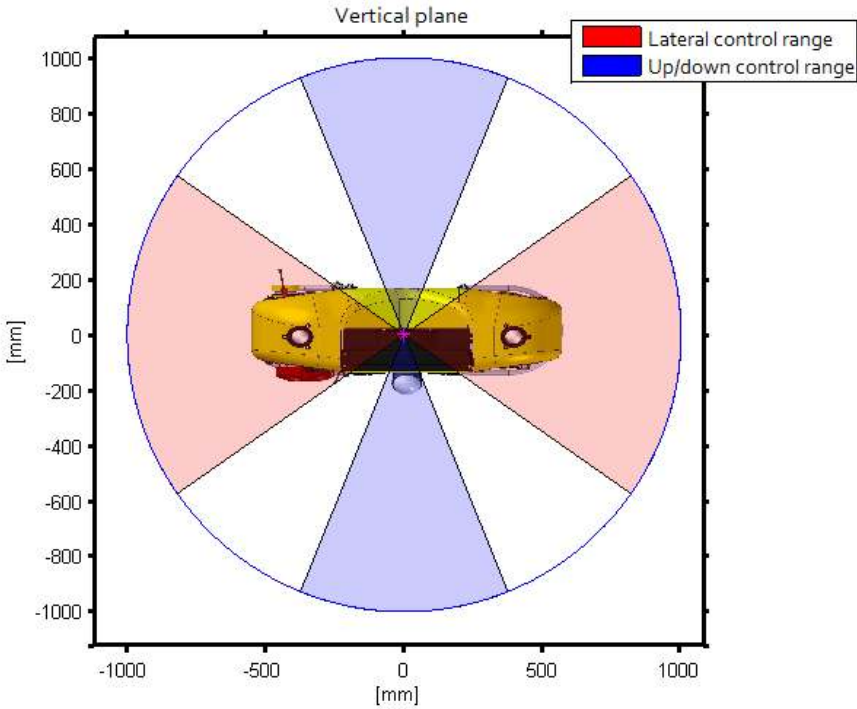


Figure 3.10. The following figure illustrates the vertical plane of the AUV. It is possible for the operator to change control zones for the vehicle. There are two areas that can be changed, one for the lateral control and one for the up/down control. Obstacles within the area are used in the control of the AUV.

3.3.2 Algorithm 2 : Heading Control

The algorithm is based on two parts, a positive and a negative heading function. The positive heading function is calculated from the desired way-point sent from the mission planner to the way-point control component, and translated to a positive utility function in the follow track component. The negative heading function is calculated from a hyperplane in the horizontal plane. The obstacles around the vehicle are projected to the hyperplane defined by a circle radius (activation circle) and a projecting range. At the final step all obstacles are summed together to a negative heading function. The negative heading function should give the vehicle a new best heading direction in the tunnel, where it is least hazardous, together with the follow track behavior. Figure 3.11 illustrates the AUV and the control zone.

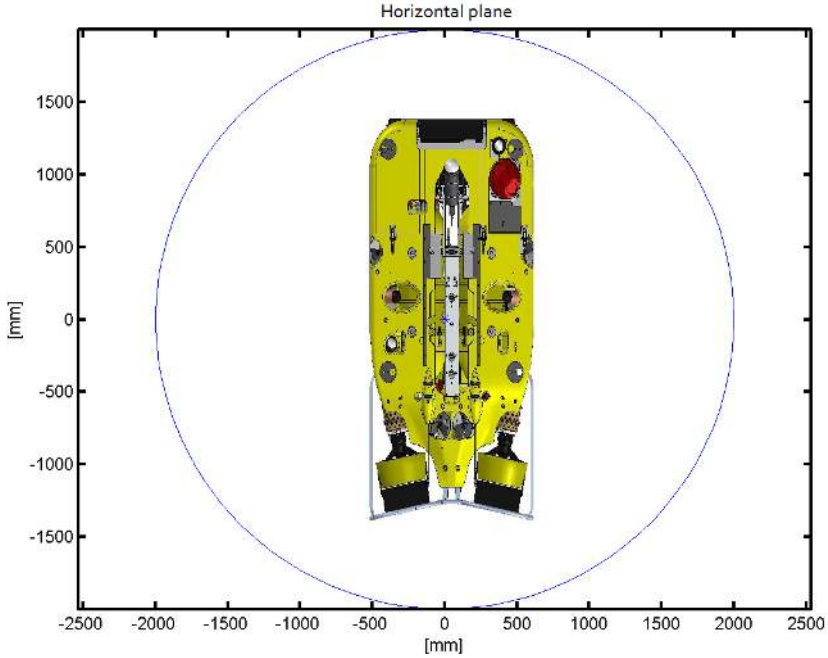


Figure 3.11. The following figure illustrates the horizontal control zone of the AUV. It is possible for the operator to change the control zone and the projecting range.

Example 3.1: Heading Control

The following example illustrates the use of algorithm 2. Figure 3.12 shows a scenario where the AUV has two obstacles at bearings of 33 degrees and 147 degrees. The obstacles are projected to the vehicle horizontal plane and two utility functions are built and sent to the arbiter for further processing. Figure 3.13 illustrates the conversion to the utility functions.

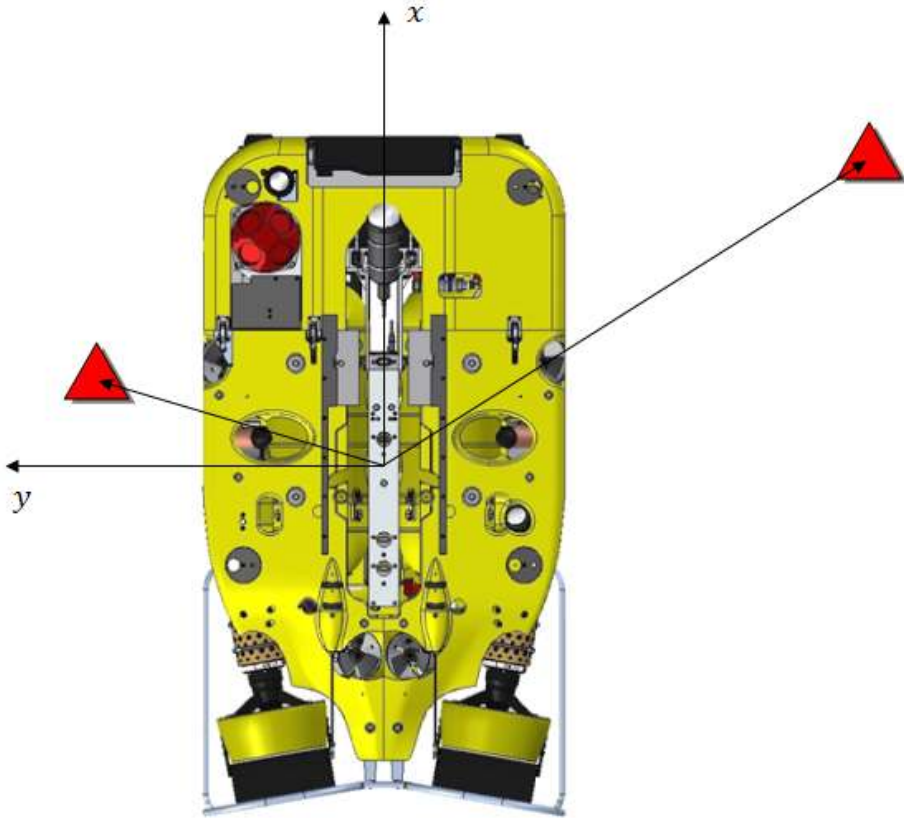


Figure 3.12. The following figure illustrates a scenario where the AUV detects two obstacles at different bearings and distances. The obstacles are projected to the vehicle horizontal plane and analyzed in algorithm 2. Figure 3.13 illustrates the transformation to the utility functions.

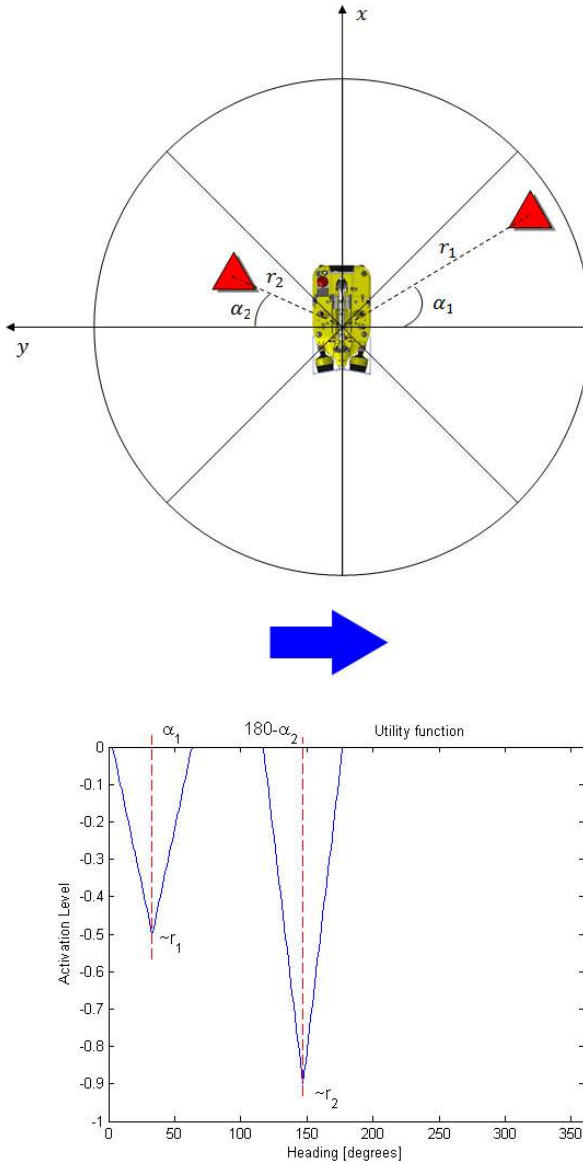


Figure 3.13. The following figure illustrates an example where two obstacles are detected and analyzed. The heading algorithm is based on a circle defined by the vehicle heading spectra ($0 \rightarrow 360$ degrees). The circle is divided into sections where an obstacle can be added. A section/area consists of a distance and an angle to an obstacle. When an area has been occupied and another want the same, the obstacle closest to the vehicle is used. The second step in the algorithm is to go through each occupied section and create a utility function for each. The utility function's center is set to the bearing (α_1 , $180 - \alpha_2$) to the obstacle and the height is proportional to the range (r_1 , r_2) to the obstacle. The spread of the utility function can be changed by the operator.

Chapter 4

Results

The following chapter presents the results based on the proposed algorithms from Chapter 3. In order to analyze and compare the algorithms, several scenarios have been developed and used for the comparison between different control settings.

4.1 Tuning Parameters

There are several settings to choose between when setting up a mission. The operator can for example choose spread on the different utility functions, priority, and of course the different control zones described in previous chapter. Table 4.1 shows the default settings used in all tests.

	Tunnel Inspection	Follow Track	Stabilize Heading
Priority	0.9	0.8	0.5
Spread X	50	100	-
Spread Y	50	100	-
Spread Z	-	-	-
Spread Heading	-	100	100

Table 4.1. Default settings used in all tests.

The above proposed settings have been decided through a series of small tests and no investigations are made on these. There are also a couple of general settings, and these are: the maximum numbers of obstacles are set to 100 (map memory), the size of an obstacle hit is set to one cubic meter and the sweep velocities of the sonars are set to 36 degrees/s. The above settings are in the range of available computation power (based on previous tests made by SUS). It is important that a mission is repeatable and therefore the simulator Elvis has been used. The remaining part of this chapter investigates how the heading spread, the control range and the projecting range affects the performance of the AUV.

4.2 Scenarios

A couple of scenarios have been developed to investigate the performance of the AUV. The following section shows the results from three different scenarios, and these are: Wall Tracking, Tunnel Tracking and Complex Mission, where two control strategies have been tested.

4.2.1 Wall Tracking

The purpose of the *wall tracking* test is to investigate how the heading spread and the control range affects the vehicle. The setup is as follow: the vehicle starts vertically against a wall and should try to reach a way-point placed on the other side of the wall. The way-point is placed 300 m away from the start point because all dynamics should be seen.

There are two tests where the heading spread is varied between 20% to 80% with an interval of 20%. The control range is set to 30 m and the projecting range to 1 m. The first test only uses the horizontal control (heading control) and the second includes also the lateral control with the same control and projecting ranges.



Figure 4.1. The above figure shows a snapshot from the wall tracking test. The AUV should try to reach a way-point behind a wall 300 m away from the start position. Two different tests are made for the comparison between the various heading spreads and control modes.

An overview of the simulation environment is illustrated in Figure 4.1. The AUV is at a minimum distance of 8 m from the tunnel wall and at a maximum distance of 30 m during the horizontal control (without the final turn). When also including the lateral control the minimum distance is at 12 m and the maximum at 30 m. Figure 4.2 and 4.3 illustrates the results.

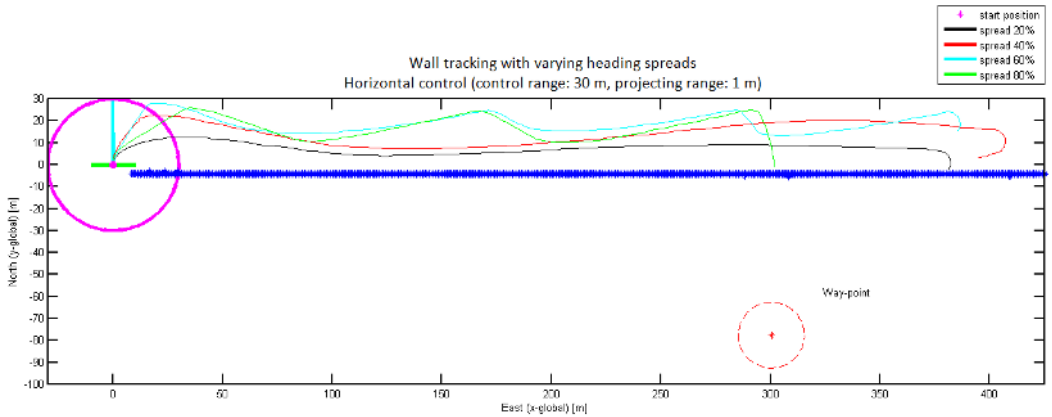


Figure 4.2. The following figure illustrates the track of the AUV during the wall tracking simulation. The purpose of the test is to investigate how good the AUV can follow a wall, and it is done by setting up a way-point in front of the vehicle and behind a wall. Only horizontal control has been used during the simulation with a control range of 30 m and a projecting range of 1 m. A higher value of the spread increases the distance to the wall and the oscillation. The minimum distance to the wall is 8 m and the maximum is 30 m and these occurs when using 20% respective 60% and 80% in spread.

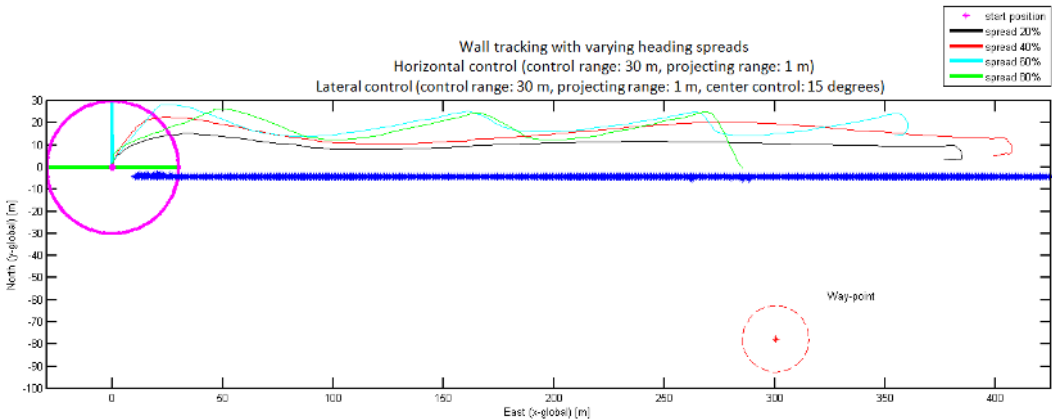


Figure 4.3. The following figure illustrates the track of the AUV during the wall tracking simulation. The purpose of the test is to investigate how good the AUV can follow a wall, and it is done by setting up a way-point in front of the vehicle and behind a wall. Horizontal control and lateral control have been used during the simulation with a control range of 30 m and a projecting range of 1 m. A higher value of the spread increases the distance to the wall and the oscillation. The minimum distance to the wall is 12 m and the maximum is 30 m and these occurs when using 20% respective 60% and 80% in spread.

Several conclusions can be made from this test. First of all, the lateral thrusters are active due to the lateral control which implies in higher energy consumption. The differences between the two strategies are not much, which can be seen in the figures. A higher spread results in more oscillation and a greater distance to the tunnel wall. The AUV managed to turn around better when using both lateral and horizontal control. There are more snapshots from the simulation in Appendix A.

4.2.2 Tunnel Tracking

The following test investigates how the heading spread and the two control strategies affects the performance of the AUV in a small tunnel. The setup is as follow: the vehicle should reach a way-point placed 700 m ahead of the AUV. During the track the AUV must handle turns and varying diameters of the tunnel. The tunnel has a maximum diameter of 15 m and a minimum of 8 m. Figure 4.4 illustrates the AUV in the tunnel tracking simulation.



Figure 4.4. The above figure shows a snapshot from the tunnel tracking test. The AUV should try to reach a way-point 700 m from the start position. The track has several turns and this figure illustrates one.

There have only been added test results with the horizontal and lateral control because only horizontal control could not manage the tunnel track. Figure 4.5 and 4.6 illustrates the results from the simulation.

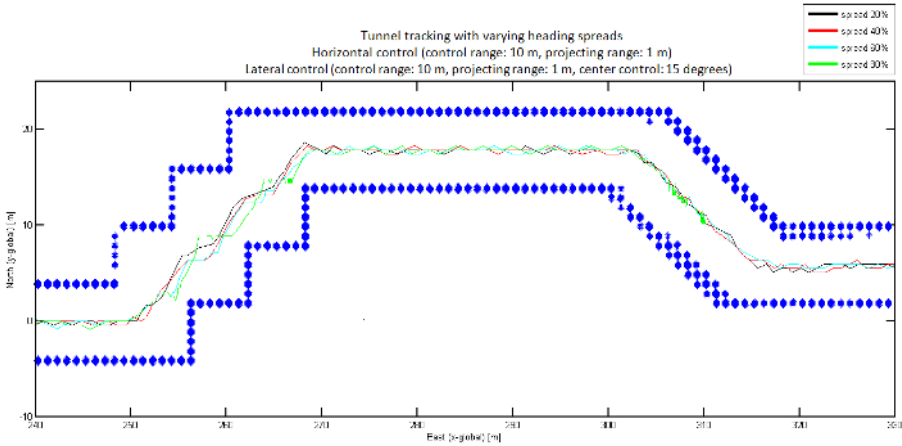


Figure 4.5. The above figure illustrates a snapshot of the track during the tunnel tracking test. Horizontal control and lateral control have been used during the simulation with a control range of 10 m and a projecting range of 1 m. The diameter of the tunnel is 8 m. The spreads 20% and 40%, managed the tunnel in the best way, i.e. the heading was in line with the tunnel. A higher value resulted in a deterioration of the heading direction and at 80% in spread the AUV could not manage the tunnel.

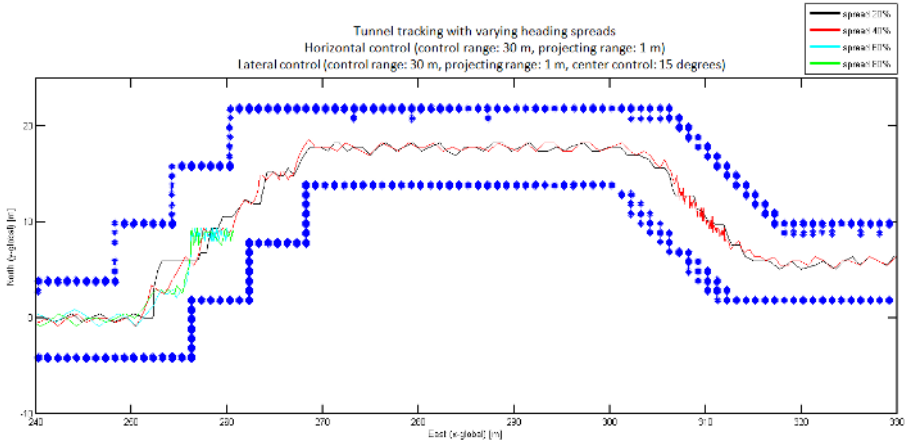


Figure 4.6. The following figure shows a snapshot of the track during the tunnel tracking test. Horizontal control and lateral control have been used during the simulation with a control range of 30 m and a projecting range of 1 m. The spread of 20% manage the tunnel in the best way, the spread of 40% managed the tunnel with several complications and a higher value did not manage the tunnel at all. The reason why the higher values did not complete the tunnel was the wrong heading direction, this can be seen at $x = 258$ m where the AUV only was moved in a lateral direction because of the east heading direction. The diameter of the tunnel is 8 m.

During the simulation the lateral thrusters were used a lot hence the lateral control. Another result that not can be seen in the figures is the heading angle of the AUV. When increasing the heading spread the direction of the AUV deteriorates. The two lower spreads, 20% and 40%, implies in a good heading direction in the tunnel. The two higher values turns the heading to an east direction, which can be seen at $x = 258$ m where the AUV only uses the lateral control.

A higher value in the lateral control range increases the oscillation of the AUV. The reason is the speed control, which is based on the control range and is proportional to the distance to the tunnel wall.

Proposed New Settings

To improve previous results a new test has been made with several modifications, and these are: decreasing the lateral control range to 3 m and increasing the projecting range to 2 m. The horizontal control range is set to 10 m with a projecting range of 2 m. The above proposed setting aims to reduce the zigzags.

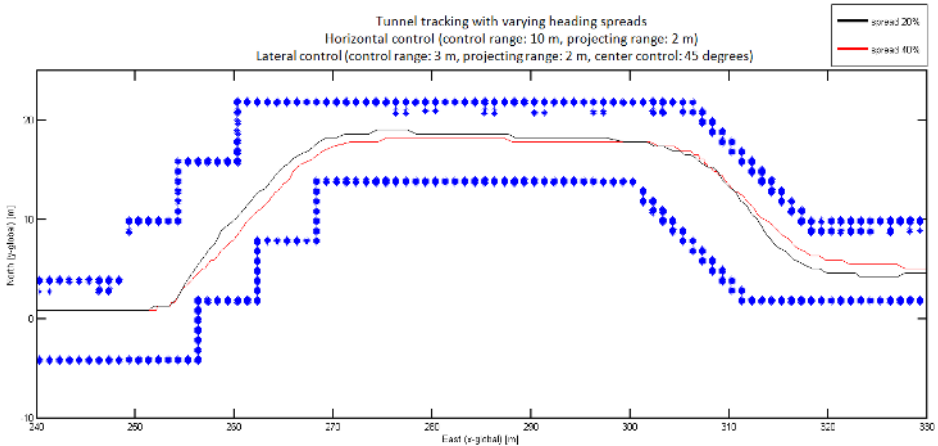


Figure 4.7. The following figure illustrates the track of the AUV during the tunnel tracking test. There have been several changes to improve the result from the previous tests. The horizontal control range is set to 10 m with a projecting range of 2 m. The lateral control range is set to 3 m with a projecting range of 2 m. The track of the AUV became smoother compared to previous tests.

Figure 4.7 shows the results with the new settings. Compared to previous tests the track of the AUV became smoother. There have only been added test results with 20% and 40% in spread because a higher value did not manage the tunnel track. There are more snapshots from the simulation in Appendix B.

4.2.3 Complex Mission

The purpose of the *complex mission* is to combine different scenarios during a mission. The track of the final scenario is approximate 2000 m and starts at position (0, 0) and ends at (50, -10). Figure 4.9 illustrates the track with nine way-points to be reached during the simulation.

The setup is as follow: A control range of 15 m and a projecting range of 2 m are used in the horizontal (heading) control. In the lateral control, a control range of 4 m and a projecting range of 3 m are used.

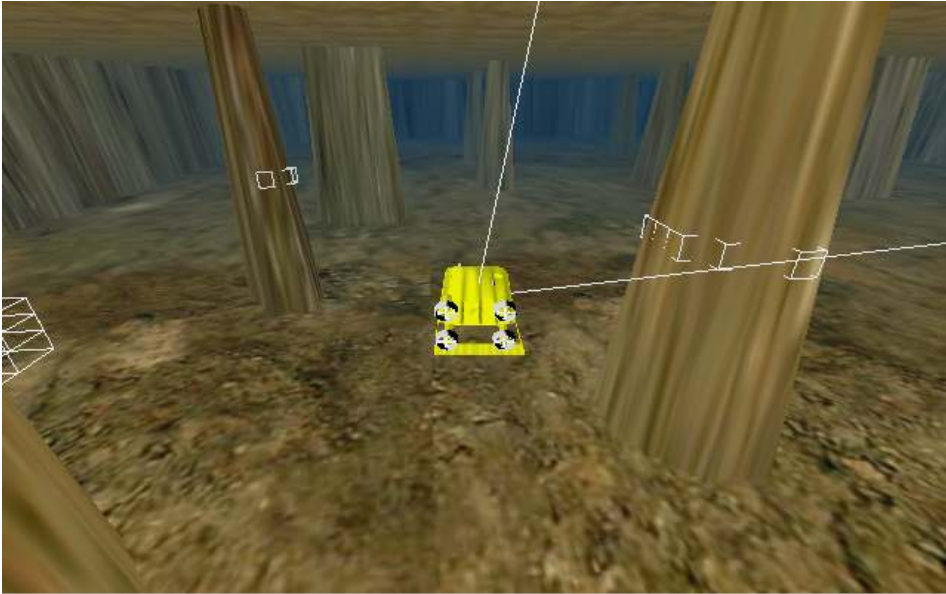


Figure 4.8. *The above figure shows a snapshot from the complex mission simulation. The AUV should in this particular scenario avoid obstacles and follow a way-point.*

Figure 4.8 illustrates the AUV in a test environment where the heading algorithm is especially tested.

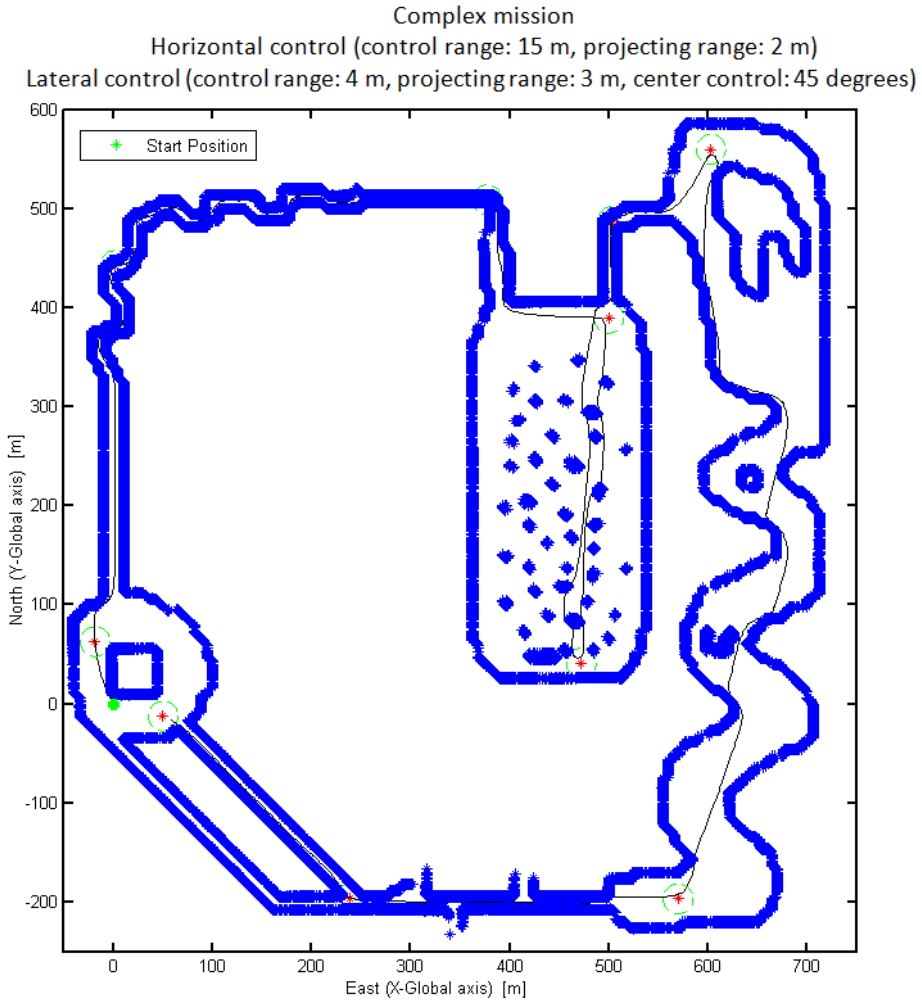


Figure 4.9. The following figure illustrates the track of the AUV during the complex mission test. The purpose of the test is to go through as many situations as possible. The track consists of nine way-points, and to reach each way-point the AUV must pass narrow places, turns, obstacles etc. The AUV starts at position $(0, 0)$ and should clockwise follow the way-points.

The AUV managed the entire track without any big problems. The proposed control settings worked good in all situations. Figure 4.10 shows a part of the track and there are more snapshots from the simulation in Appendix C.

Complex mission
Horizontal control (control range: 15 m, projecting range: 2 m)
Lateral control (control range: 4 m, projecting range: 3 m, center control: 45 degrees)

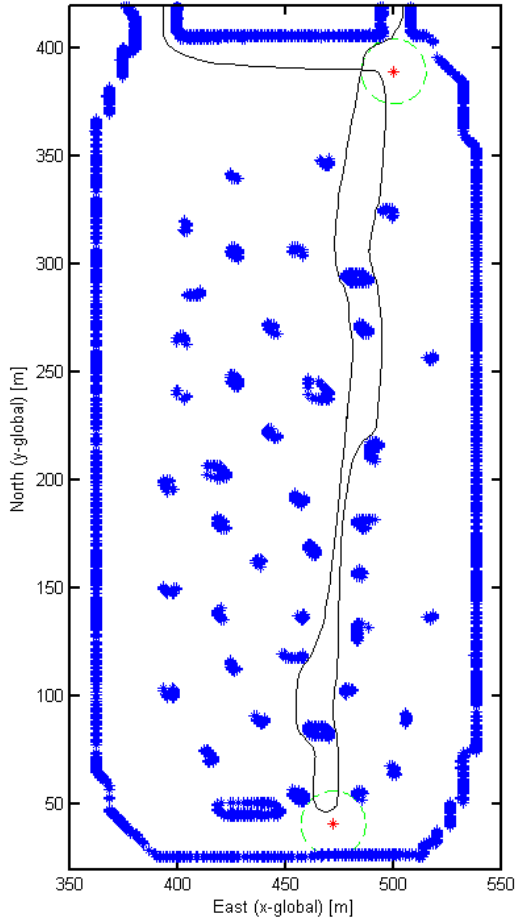


Figure 4.10. The following figure illustrates a snapshot of the track during the complex mission simulation. In this situation the AUV should pass an area with several obstacles and reach a way-point and then go back. The minimum distance to an obstacle is 3 m.

Chapter 5

Summary

The following chapter summarizes the results from the previous chapters and proposes further developments.

5.1 Conclusions

The implementation of the behavior-based control strategy has been successful. A behavior-based control strategy is sufficient to manage tunnel assignments. The implemented behaviors provide the ability to avoid obstacles, maintain a proper distance to the tunnel wall, and follow desired way-points.

Only simulations with simulated data have been made, and therefore real measurements and tests are necessary before using the implemented control system in real situations. The tuning of the control parameters should be considered and maybe configured for each mission. Based on the results from Chapter 4 there can be made some conclusions:

- Lateral and horizontal control are recommended to be used in tunnel assignments. Only horizontal control requires well defined control parameters. The use of both control modes decreases the sensitivity of the control parameters.
- All tests confirms that the heading spread parameter in the horizontal control should be in the range of 20% to 40%. General a higher spread deteriorates the heading of the AUV and especially in a small tunnel environment. It is also easier to detect tunnel branches if the spread is small.
- The control range parameter in the proposed algorithms should be configured for each mission. It is recommended to use a lateral control range smaller than the diameter of the tunnel due to the activation of the side thrusters. This will decrease the battery consumption.

- The control range parameter should be seen as a safety parameter. If the vehicle should be at least 2 m from the tunnel wall, a lateral control range of 3 m can be a good choice. The horizontal control is also dependent of the mission. If the operator wants a high forward velocity, a greater control zone may be used because of the dynamics of the AUV.
- The projecting range parameter, in the horizontal and vertical plane, should be seen as a looking forward parameter. It is important to not look too far because the obstacles are projected to each plane and therefore the uncertainty is proportional to the distance in some sense.

5.2 Further Developments

The following section proposes further developments of the implemented components:

The implemented coordinate system has singularities at $\theta = \pm 90$ degrees which restricts the motion of the AUV. A new angle definition is required to solve this and one solution is to use Quaternions [18].

There have only been investigations of an AUV with a pitch and a roll angle equal to zero. The next step is to allow these two degrees of freedom to be nonzero. There may for example be situations when the AUV should follow a tunnel upwards or downwards.

The up and down control of the vehicle has not yet been investigated, and to do that a new simulation environment is required where the water depth changes.

The algorithm 1 can be improved by allow more than one obstacle to be part in the calculation of the control signal. One improvement would for example be to take the closest obstacle from each side of the AUV, take the difference between these and then calculate the lateral control signal.

There are a lot of settings to choose between when setting up a mission. There have for example been investigations of how the priority, the spread, the control and projecting ranges affect the performance of the AUV. Another issue is to change the form of the utility function, an example is to use a Gaussian appearance and compare the results with the previous form.

The tunnel inspection component should in my opinion be separated into two components, and these are: one component for controlling the AUV to a desired distance from the tunnel wall and another merged with the avoid obstacle behavior. The reason for a separation is that a tunnel inspection mission has many definitions and therefore it is impossible to define a general tunnel inspection component. It is supposed that the operator should define a mission by activating desired components/behaviors with appropriate settings.

The developed system is based on data from two range sonars. The sonar data is sent to the occupancy grid for building the world around the AUV. A development of the structure would be to integrate other available sensors such as doppler-logs and cameras. By combining several sensors the estimation of the environment can be better.

All available settings for a mission should be integrated into one application and therefore some functionality in the GUI should be transferred to the mission planner. It should at least be possible to choose desired control ranges. Other settings such as the size of the boxes and the number of boxes in the memory can be investigated in the simulator and then hard-coded.

Last but not least, the tuning of the low level controller (PID) affects the BBC strategy and the performance of the AUV. This must be taken into account when evaluating the results. I think the hardest task is to set up the different way-points before a mission; this places a great responsibility on the operator.

Bibliography

- [1] A. Broggi, M. Bertozzi, A. Fascioli, G. Conte. "AUTOMATIC VEHICLE GUIDANCE, The Experience of the ARGO Autonomous Vehicle". ISBN: 981-02-3720-0. World Scientific Co. Publisher, Singapore, April 1999.
- [2] Alberto Elfes. "Using occupancy grids for mobile robot perception and navigation". *IEEE Computer*, pages 46-57, June 1989.
- [3] Hans Moravec and Alberto Elfes. "High resolution maps from wide angle sonar". In *Proc. IEEE Int. Conf. Robotics and Automation*, 116-121.
- [4] Marc Carreras Pérez, Ph.D. "A Proposal of a Behavior-based Control Architecture with Reinforcement Learning for an Autonomous Underwater Robot". Girona University, May 2003.
- [5] Michael R. Benjamin, Ph.D. Abstract of "Interval Programming: A Multi-Objective Optimization Model for Autonomous Vehicle Control". Brown University, May 2002.
- [6] Pushkin Kachroo and Sabiha Wadoo. "Autonomous Underwater Vehicles: Modeling, Control Design and Simulation". ISBN: 978-1-4398-1831-2, CRC Press 2010.
- [7] R. Brooks. "A Robust Layered Control System for a Mobile Robot". *Journal of Robotics and Automation*, vol. RA-2, no. 1, pp. 14-23, April 1986.
- [8] Ronald C. Arkin. "Behavior-Based Robotics". MIT Press, Cambridge, MA, 1998, ISBN: 0262011654.
- [9] Steven J. Rasmussen and Tal Shima. "UAV Cooperative Decision and Control". ISBN: 9780898716641. Society for Industrial and Applied Mathematics, U.S. Publisher, February 2009.
- [10] T.I Fossen. "Guidance and Control of Ocean Vehicles". ISBN 0-471-94113-1. John Wiley and Sons Ltd, UK, 1994.
- [11] General information about saab underwater systems. URL: <http://www.saabgroup.com/> and <http://www.seaeye.com/index.html>, February 2011.

-
- [12] General information about the company, hibbard inshore. URL: <http://www.hibbardinshore.com>, April 2011.
- [13] General information about the double eagle system. URL: <http://www.seaeye.com/doubleeagle.html>, June 2011.
- [14] General information about the matlab engine. URL: http://www.mathworks.com/help/techdoc/matlab_external/f29148.html, April 2011.
- [15] General information about the oceanographic systems lab. URL: <http://www.whoi.edu/page.do?pid=29862>, April 2011.
- [16] General information about the sea owl system. URL: <http://www.seaeye.com/seaowlsubrov.html>, June 2011.
- [17] Wikipedia, an introduction of euler angles. URL: http://en.wikipedia.org/wiki/Euler_angles, July 2011.
- [18] Wikipedia, an introduction of quaternions. URL: <http://en.wikipedia.org/wiki/Quaternion>, July 2011.
- [19] Wikipedia, an introduction of sonars. URL: <http://en.wikipedia.org/wiki/Sonar>, April 2011.
- [20] Wikipedia, an introduction of xml. URL: <http://en.wikipedia.org/wiki/XML>, July 2011.
- [21] Wikipedia, general information about the google driverless car. URL: http://en.wikipedia.org/wiki/Google_driverless_car, August 2011.
- [22] Wikipedia, general information about the us military darpa grand challenge. URL: http://en.wikipedia.org/wiki/DARPA_Grand_Challenge, August 2011.

Appendix A

Wall Tracking

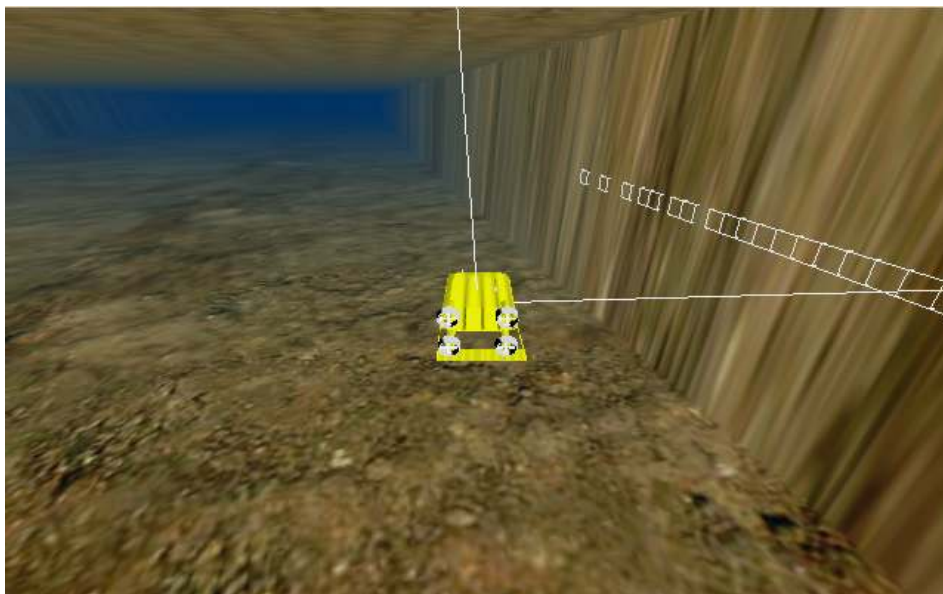


Figure A.1. The above figure shows a snapshot from the wall tracking test. The AUV should try to reach a way-point behind a wall 300 m away from the start position. Figure A.2 and A.3 presents the position and the utility functions at this specific time instance. Figure A.4 and A.5 illustrates two test results.

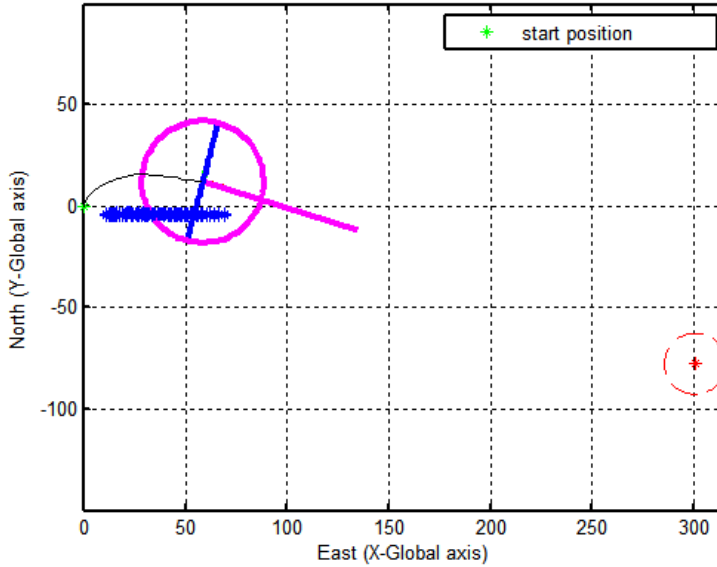


Figure A.2. The following figure illustrates an instance in the wall tracking test. The figure shows the position and the track of the AUV, and is associated with Figure A.1 and A.3.

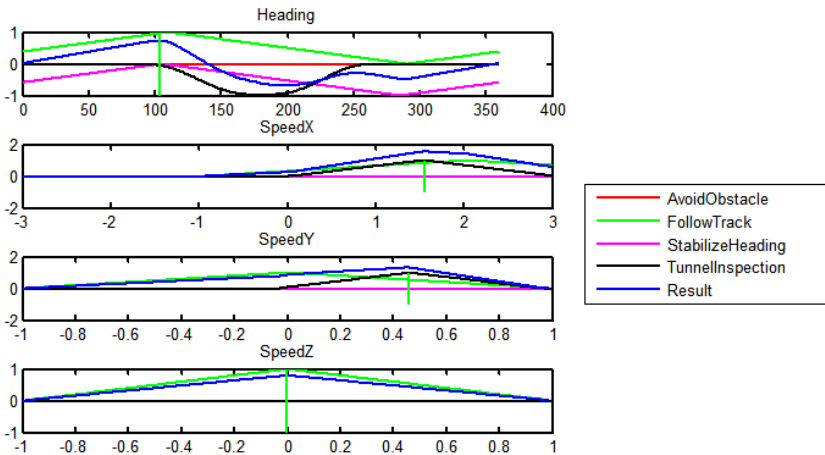


Figure A.3. A snapshot of the utility functions associated with Figure A.1 and A.2. As can be seen there is a negative utility in the tunnel inspection heading (black line) between 100 and 250 degrees because of the tunnel wall. The port velocity (SpeedY) is nonzero because there are obstacles within the lateral control zone. The heading output is approximate 100 degrees, the forward speed is 1.5 knots and the lateral speed is approximate 0.5 knots.

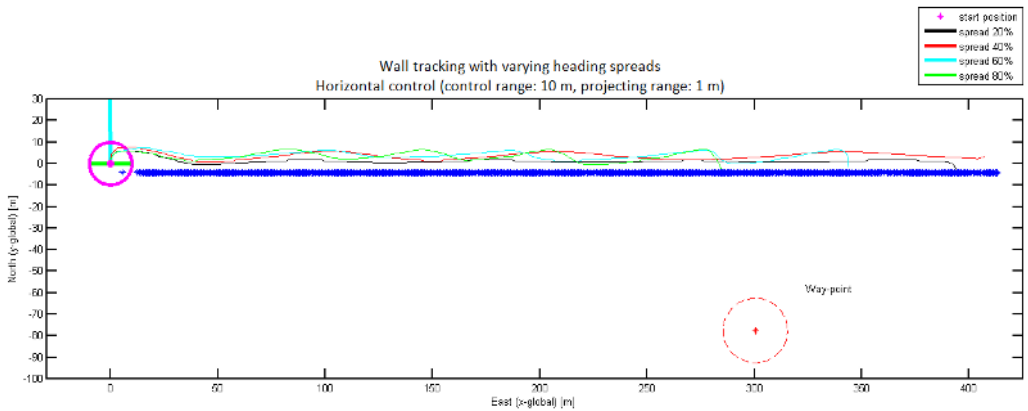


Figure A.4. The following figure illustrates the track of the AUV during the wall tracking simulation. The purpose of the test is to investigate how good the AUV can follow a wall, and it is done by setting up a way-point in front of the vehicle and behind a wall. Only horizontal control has been used during the simulation with a control range of 10 m and a projecting range of 1 m. A higher value of the spread increases the distance to the wall and the oscillation. The minimum distance to the wall is 3.5 m and the maximum is 12 m and these occurs when using 20% respective 60% and 80% in spread.

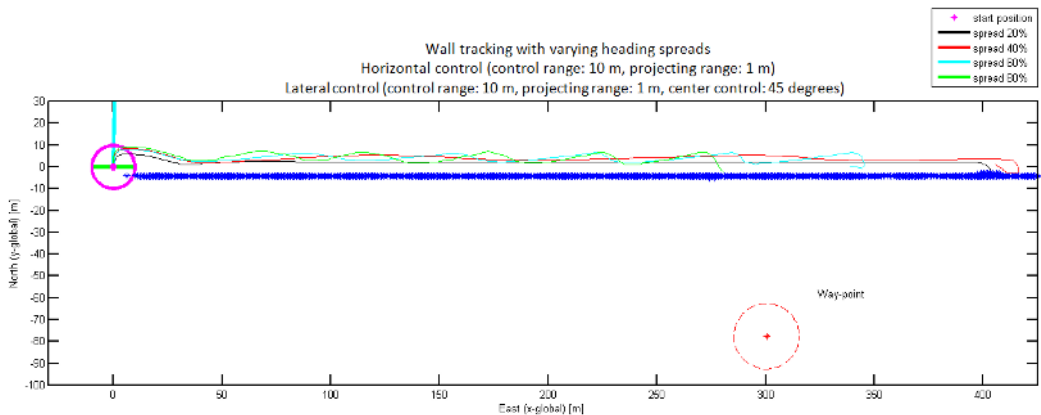


Figure A.5. The above figure shows the track of the AUV during the wall tracking simulation. The purpose of the test is to investigate how good the AUV can follow a wall, and it is done by setting up a way-point in front of the vehicle and behind a wall. Horizontal control and lateral control have been used during the simulation with a control range of 10 m and a projecting range of 1 m. A higher value of the spread increases the distance to the wall and the oscillation. The minimum distance to the wall is 5 m and the maximum is 13 m and these occurs when using 20% respective 60% and 80% in spread.

Appendix B

Tunnel Tracking



Figure B.1. The above figure shows a snapshot from the tunnel tracking test. The AUV should try to reach a way-point 700 m from the start position. The track has several turns and this figure illustrates one. Figure B.2 and B.3 illustrates the position and the utility functions at this specific time instance.

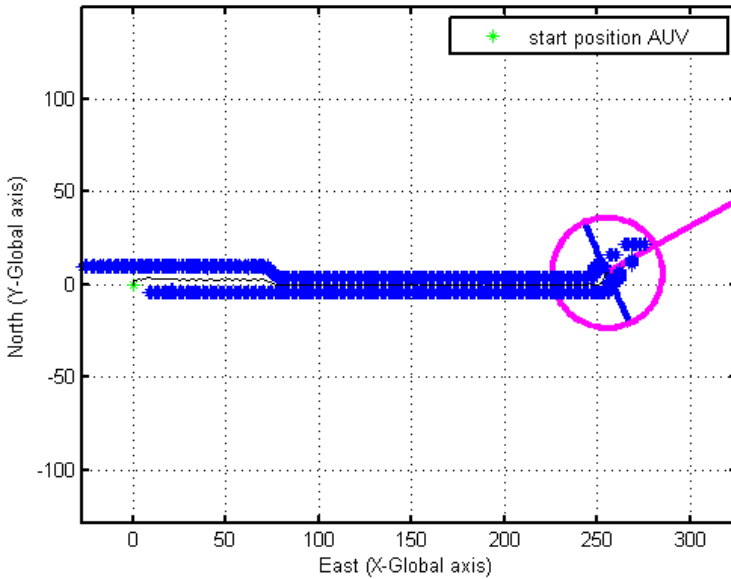


Figure B.2. The following figure presents the position and the track of the AUV during the tunnel tracking test. The AUV should manage a turn with the following control settings: lateral control and horizontal control with a control range of 30 m and a projecting range of 1 m. Figure B.3 shows the utility functions at this time instance.

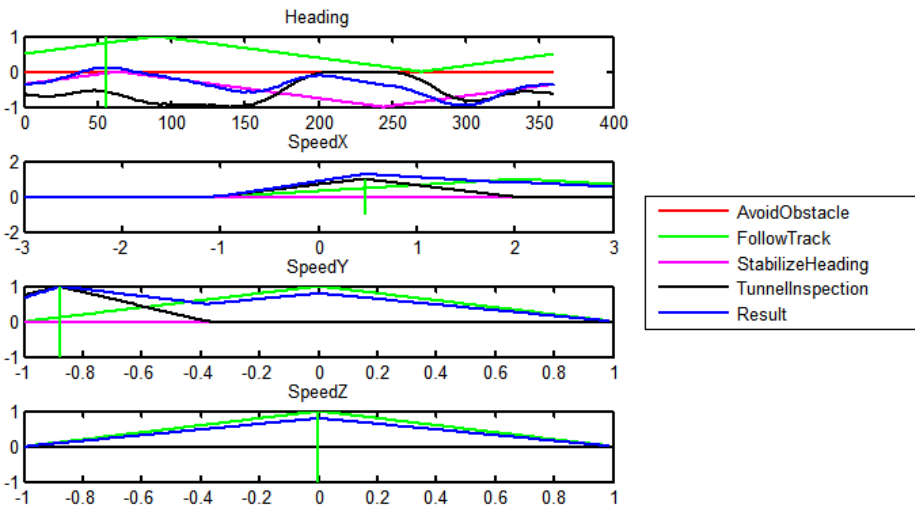


Figure B.3. The above figure shows a snapshot of the utility functions during the tunnel tracking test. The heading output is approximate 60 degrees, the forward speed is 0.5 knots and the lateral speed is approximate 0.9 knots.

Appendix C

Complex Mission

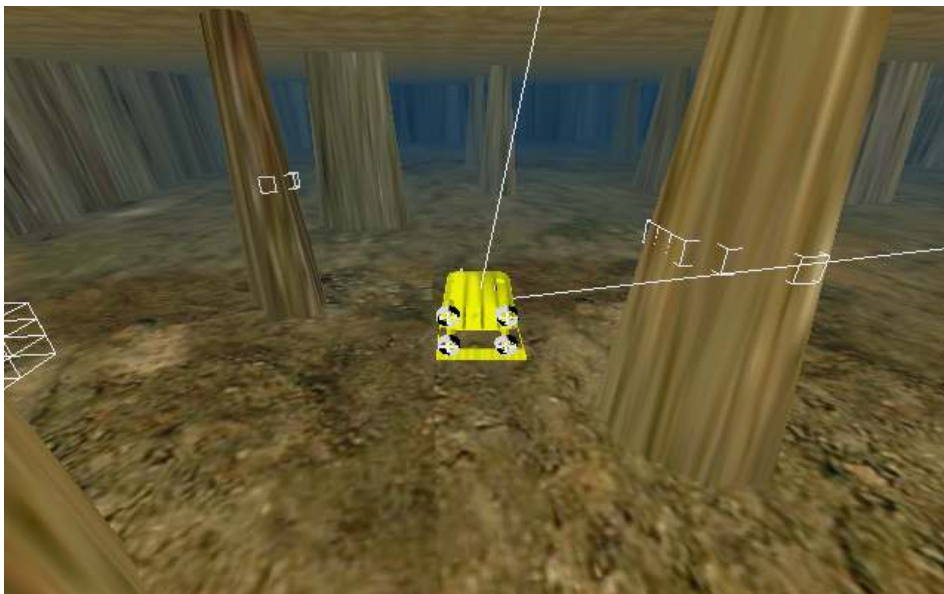


Figure C.1. The following figure shows a snapshot from the complex mission simulation. The AUV should in this particular scenario avoid obstacles and follow a way-point. Figure C.2 and C.3 illustrates the position of the AUV and the utility functions at this time instance.

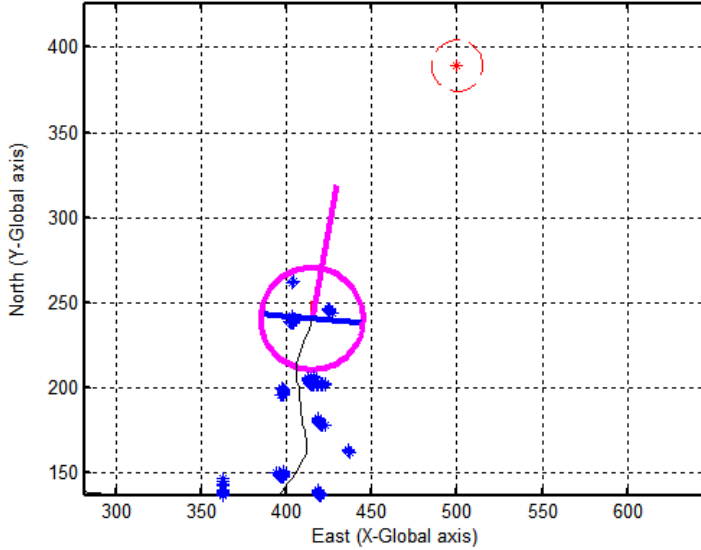


Figure C.2. The following figure illustrates the position and the track of the AUV at a specific time instance. As can be seen there is one obstacle at each side of the AUV which implies in two negative humps in the tunnel inspection's utility function. Figure C.3 shows the different utility functions.

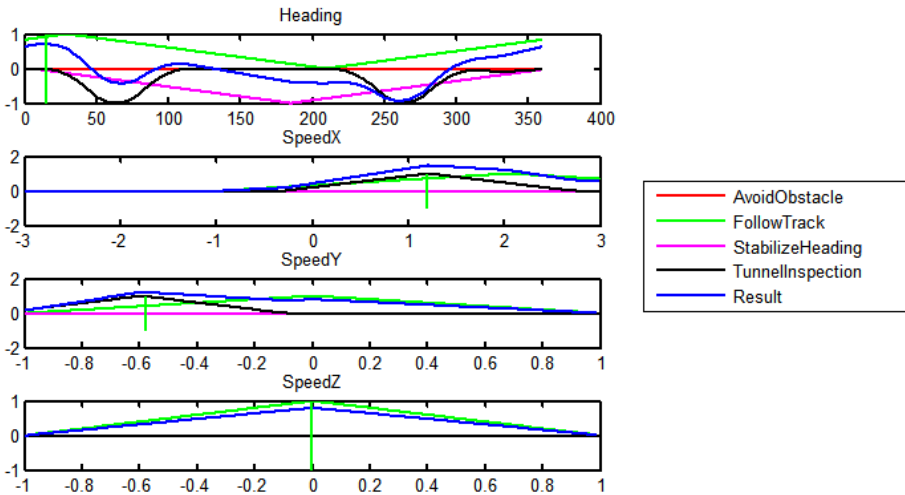


Figure C.3. The above figure presents a snapshot from the utility functions during the complex mission test. Worth noting are the two negative humps in the tunnel inspection's utility function which indicates in danger at both sides of the AUV.

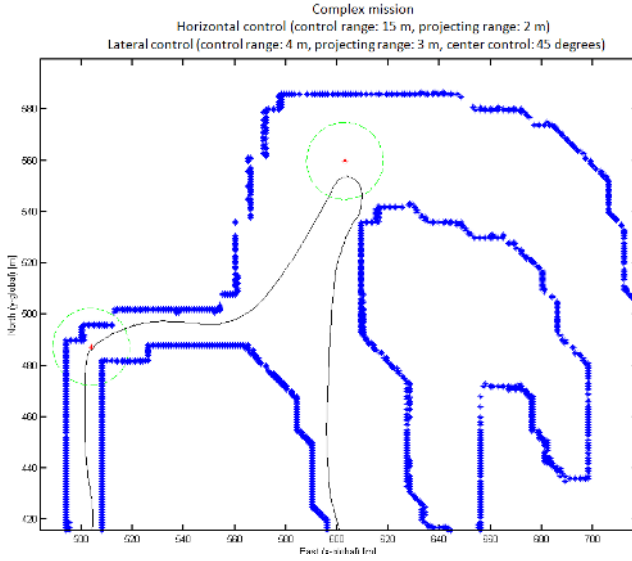


Figure C.4. The following figure illustrates a snapshot from the complex mission simulation. The AUV have reached two way-points and is on the way to the next. The AUV is traveling from the left to the right.

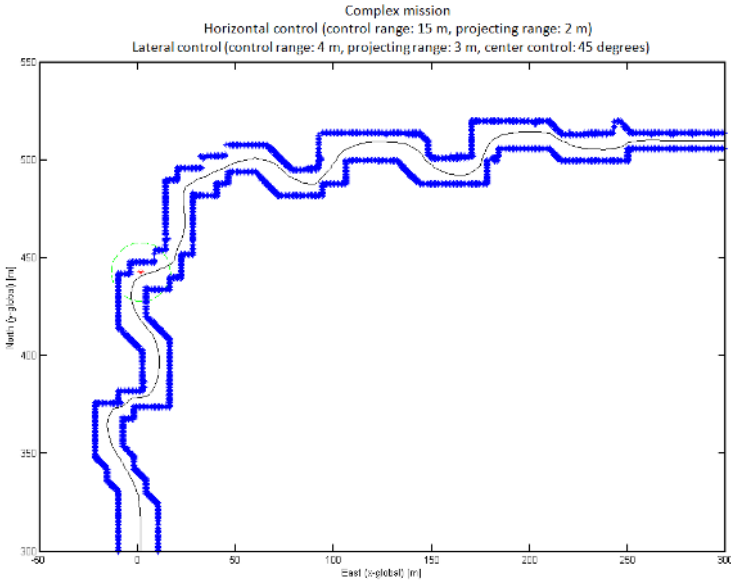


Figure C.5. The above figure shows a snapshot from the complex mission test. This scenario illustrates a tunnel with several turns. The diameter of the tunnel is minimum 8 m and the AUV passes from the bottom to the top.

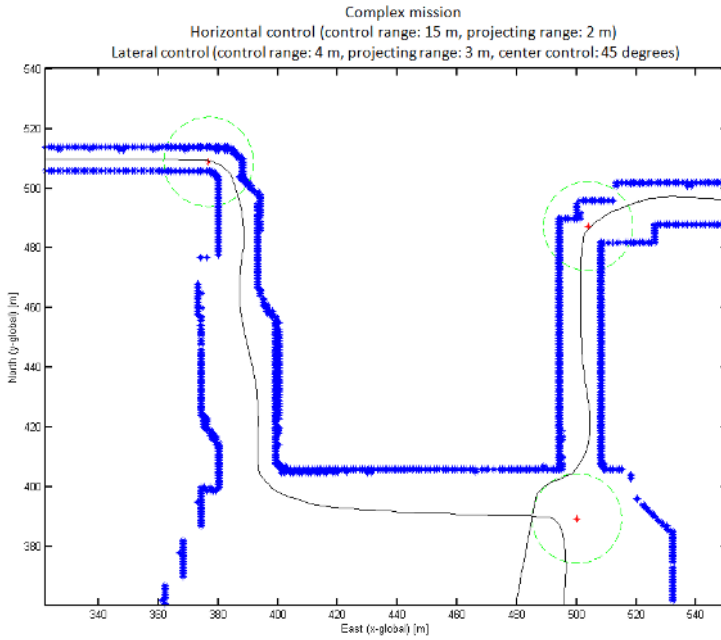


Figure C.6. The following figure presents a snapshot from the complex mission test.

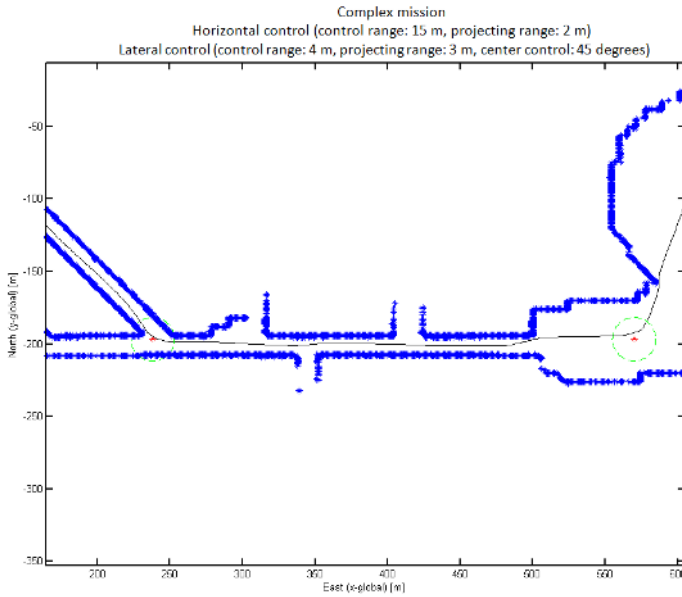


Figure C.7. The above figure illustrates a snapshot from the complex mission simulation.