INSTITUTE OF SOUND AND VIBRATION RESEARCH
UNIVERSITY OF SOUTHAMPTON

MASTER OF PHILOSOPHY

(MPHIL)

ROBUST CONTROL
OF
AN UNDERWATER VEHICLE

Nor Suzlinda Mohd. Saleh

OCTOBER 2003

INSTITUTE OF SOUND AND VIBRATION RESEARCH
UNIVERSITY OF SOUTHAMPTON

ABSTRACT

MASTER OF PHILOSOPHY (MPHIL)

ROBUST CONTROL OF AN UNDERWATER VEHICLE

By

Nor Suzlinda Mohd. Saleh

An H-infinity controller has been designed for the heading control of an underwater vehicle. The performance weighting function, $W_1(s)$ and the robustness weighting function, $W_3(s)$ were obtained from experimental data. The results of simulated heading responses were determined using an underwater vehicle simulation program, *AutoROV* and MATLAB robust control toolbox. A nonlinear simulation program was also used to observe the behaviour of the underwater vehicle to nonlinear underwater dynamics. The simulated results of the H-infinity controller were compared with the PID controller and results showed that the H-infinity based controller can withstand uncertainties such as random noise and rudder delay better than a PID based controller.

Table of Contents

Chapter 1

Introduction

The first section briefly introduces a short history of underwater vehicles and then concentrates on the autonomous underwater vehicle (AUV). The background research on the current project is then discussed in the second part of this section.

The earth consists of the land, the space and the water area. Spacecrafts or spaceships are used to investigate unfamiliar objects in space. On the other hand, underwater vehicles such as submarines and unmanned underwater vehicles are used for exploration as well as for other survey work in the underwater world.

Underwater vehicles have existed for many years. Before the 1960s, the underwater vehicle was mainly used for military purposes, but it is now shifting towards scientific and industrial tasks such as studying marine animals and repairing underwater cables. By having such vehicles, it is possible for humans to expand their knowledge to places which are alien to human beings. In addition, these vehicles can lower human risks to explore beneath 70 % of the earth's surface area, the underwater world.

1.1 Underwater Vehicles

Generally, there are two types of underwater vehicle: the remotely operated vehicle (ROV) and the autonomous underwater vehicle (AUV). Such vehicles are also sometimes known as unmanned underwater vehicles (UUVs). ROVs have been developed much earlier compared to AUVs and began to mature during the early 1980s. Whereas, AUVs have only been recognized for the last 30 years. AUVs were initially either large or expensive. These problems are gradually being eliminated with developments in underwater technology. The differences between ROV's and AUV's are shown in Table 1.1.

ROV	AUV
1) has a cable for communication link and power	1) untethered
2) has a support ship	2) no support ship; self contained
3) uses multiple thrusters	3) uses a propulsion unit and control surfaces
4) boxed typed or bluff shape	4) streamlined shape to reduce drag effect
5) do not need to carry maximum energy due to 1)	5) limited energy
6) can hover due to 3)	6) move forward to manoeuvre
7) limited applications and mostly are for oil industry	7) wider applications since it can travel deeper into the water; extended to oceanographic and under ice surveying

Table 1.1: Differences between ROV's and AUV's

ROVs have a support ship or deployed from oilrig or harbour as well as a cable for communication link and power supply which limits the depth of the underwater vehicle. In addition, drag is introduced by the cable which slows down the vehicle and decrease its performance. AUVs do not have these limitations. They can go deeper into the ocean and has made AUVs more demanding in surveying work such as diamond mining and under-ice surveys; as well as in exploration expeditions. ROVs are still in use especially in the oil industry in areas such as laying cable and repairing oil pipes. However, AUVs are widening considerably the possibilities of underwater exploitation. AUVs are discussed further in the next section. Figure 1.1 and Figure 1.2 show pictures of a ROV and an AUV, respectively.

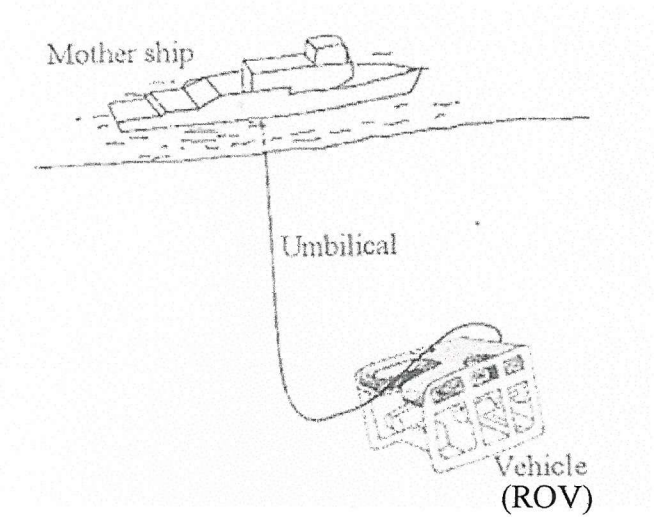


Figure 1.1: An ROV with its mother (support) ship. Taken from Lea [1].

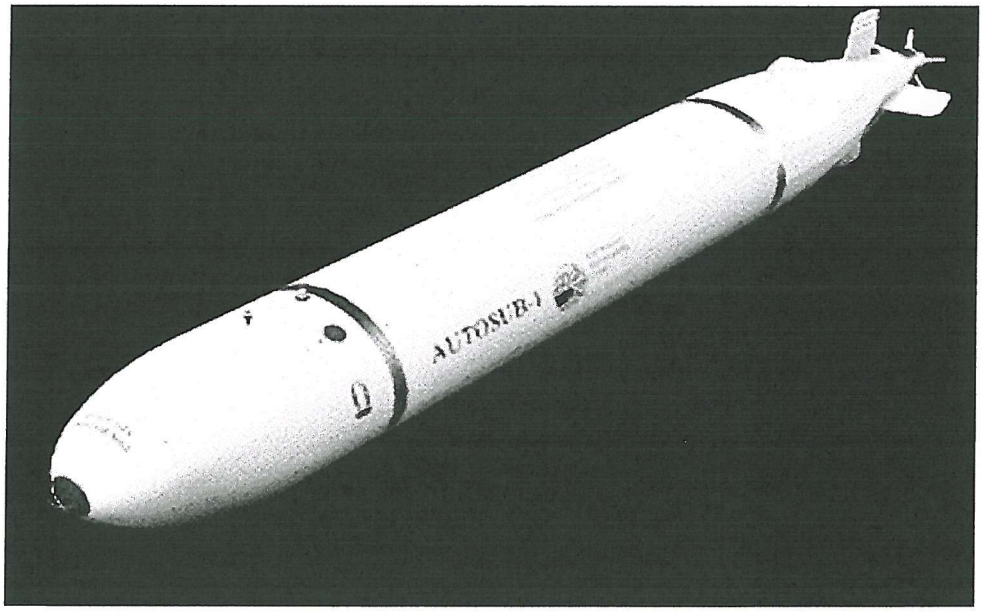


Figure 1.2: Autosub-1, an AUV developed by the Southampton Oceanography Centre.

1.1.1 Autonomous Underwater Vehicles (AUVs)

The development of AUVs has been relatively slow due their high level of complexity in technology and high risks for development. Before, the AUVs were used to test the components of the AUVs that would be used later in an AUV. But now it is being used to perform specific task such as search and survey works [16].

History and development

In the 1980s, the military community spent hundreds of millions of dollars to develop an AUV for a stealth reconnaissance platform [16]. Other companies such as Lockheed Martin's Perry Technologies and the Mobile Undersea Systems Test (MUST) Laboratory, had also built their own AUVs for the same purpose. The AUVs built were large in size. On the other hand, a couple of academic organisations, Massachusetts Institute of Technology (MIT) Sea Grant and Florida Atlantic University (FAU), began to develop smaller, low-cost AUVs. This is because trials can be made more cost effective which speeds up the development of such vehicle. As a result, in 1994, the Office of Naval Research (ONR) was attracted to use AUVs for ocean research work and funded these academic institutions. This development made big defense contractors realised that small AUVs can also be used in areas such as disposable mine hunting missions and surveys. Applications in the oil and gas industries are also increasing due to the resulting low total costs for a mission or survey.

The early AUVs and the companies that developed the AUVs are shown in Table 1.2 below.

Company/ Developer	Vehicle	Application
Marine Systems Laboratory of New Hampshire	*EAVE III*	Testing control system and algorithms
The Mobile Undersea Systems Test	*MUST*	Testing advanced systems such as the propulsion unit for the Remote Minehunting System (RMS)
Defense Advanced Research Project Agency (DARPA)	*DARPA*	Navy specific military missions
Florida Atlantic University (FAU) and Lockheed Martin's Perry Technologies	*Ocean Voyager I* and *Ocean Voyager II*	Test CHIRP side scan and sub-bottom sonars, video cameras
International Submarine Engineering (ISE)	*ARCS*	Under-ice survey
Applied Remote Technology (ART)	*XP-21*	Military
Monterey Bay Aquarium Research Institute (MBARI)	*Otter*	For optical system: to image the sea floor and to perform video mosaicing.
Institute of Industrial Science , Tokyo University – sponsored by Toyota Motor Corporation	*Twin Burgers*	For software development

Table 1.2: Early test bed AUVs

As time went by, the applications for AUVs widened into search works [17]. AUVs are increasingly being used in survey work because they can cover larger areas in a relatively short period of time. Table 1.3 shows the AUVs which were involved in search and survey fields.

Company/ Developer	Vehicle	Application
US Navy's Advanced Unmanned Search System	*AUSS*	To collect sonar and optical images of ocean bottom (20 000 feet)
Cooperation of Statoil, Norwegian Defense Research Establishment (FFI), Norwegian Underwater Intervention AS (NUI) and Kongsberg Simrad AS (KS)	*HUGIN*	High resolution seabed mapping
Maridan of Denmark	*MARTIN (MARIDAN)*	For oceanographic and commercial surveys such as pipeline, cable, pre-construction and bathymetric surveys.

Table 1.3: AUVs used in search and survey work

Today, there are three main areas for AUV technology: *Survey work*, *Military* and *Scientific* [18]. Among these three, survey work is currently dominating the market. There are few reports and analyses on the advantages of using AUVs for survey work. It is reported in [18] that the number of subsea installations will be doubled in 5 years time because it is believed that an AUV can cover deeper and larger areas underwater. The C&C Technologies, Inc, analysed that there was a saving of 59% in the total cost of a deepwater survey when a comparison between a deep towed system and an AUV was made. In addition, the U.S Navy had made a similar conclusion where the full ocean depth survey time was reduced when its Advanced Unmanned Search System (*AUSS*) AUV was used.

Lea [1] mentioned in his thesis that in 1994, the running costs to operate a support vessel a day was about £13,000. Therefore, if an AUV is used instead of a towed array or a ROV, the running cost for a support ship is reduced. The application of AUVs in military applications has been under development for decades because it was

expensive to run a mission. The scientific application for AUVs is increasing as more international academic and research organizations realise the usefulness of this technology. Smaller and more economical vehicles are also used for scientific survey.

Current development of the AUV technology

Currently, in the offshore survey field, *MARIDAN*, *HUGIN* and *AQUA EXPLORER 2* are the leading autonomous underwater vehicles [18]. *MARIDAN 600* is capable of diving up to 600 meters deep and it is going to be used for diamond mining surveys by De Beers. The next model *MARIDAN 3500* is going to be used for deepwater surveys. Norway's *HUGIN* AUV is being used for pipeline route surveying. The recent *HUGIN 3000* is capable of diving to 3000 meters depth. Japan's *Aqua Explorer 2* AUV has recently completed a survey to bury cables in the Taiwan Strait. Other AUVs that are also in the offshore survey industry are *Sea Oracle* by Racal Survey Group Ltd and Bluefin Robotics', *Explorer* developed by ISE, Port Coquitlam, BC, Canada, *Thesus* by ISE and *MUST* by Perry Technologies. Examples of smaller AUVs are *CETUS II* from Lockheed Martin and *REMUS* by Woods Hole Oceanographic Institutions (WHOI).

Autosub developed by the Southampton Oceanography Centre (SOC) is leading other AUVs in scientific missions [18]. In scientific research [18], the AUVs are used in collecting oceanographic data, to study the nature and life of marine animals, for searching lost objects from wrecked ships and in under-ice surveys. Another AUV which is being used in the science field is the *Urashima* AUV developed by JAMSTEC of Japan which is capable of 3500 meters depth. *Autosub-1* and *ALTEX* have been through the under-ice surveys successfully.

AUVs come in different shapes such as the 'torpedo' (Natural Environment Research Council (NERC) *Autosub*) and 'flatfish' (*NPS-AUV II*). The shapes and sizes are designed according to the tasks required. Currently, smaller sized AUVs are of increasing interest. Schexnayder *et al* [13] suggested that smaller AUVs have a

reduced power consumption. However, due to the smaller size and lighter weight, the vehicle needs to withstand greater wave disturbances.

The AUVs are also used in petrochemical industry, in petroleum, mining and communications industries. The tasks include inspecting cables and oil pipelines as well as underwater constructions. Recent applications such as the coastal and environmental management include monitoring waste, natural hazards, environmental changes and bio-diversity. Furthermore, an AUV can be used by the chemical industry for monitoring and detecting dissolved chemicals including oxygen and nitrate ions.

Future of AUVs technology

From the above discussion, the advantages of an AUVs can be summarised as below.

- Survey can be carried out to a deeper depth and larger areas can be covered. This is because no cable is attached to the vehicle. In addition, the time of a mission can be reduced.
- The total costs of survey work can be reduced.
- The understanding of ocean life and environment can be increased, especially at the deeper ocean (sea) bottom.
- In technology, the combination of advanced and miniaturised sensors can be tested on AUVs.

Even though an AUV technology gives some advantages over its 'cousin' ROV, it is still at the early stage of acceptance. This is because the technology and costs of developing an AUV are high [19]. In turn, developers are afraid of losing an AUV because it is expensive to build a replacement. Other limitations for AUVs technology are high energy storage, more computational power and accurate navigation [19]. With technology, the limitations above can gradually be reduced. Despite these drawbacks, AUVs have big potential and therefore funding to develop and to maintain those vehicles are still required.

1.2 Control

One of the possibilities to improve the AUV technology is by having a robust underwater vehicle. In addition, most of the AUVs consists of expensive sensors to provide data such as speed, on the vehicle. This makes it an expensive vehicle especially as a testing vehicle. By employing a robust controller into the vehicle, the probability of losing an expensive AUV can be reduced. To achieve this, high performance and stable controllers are required especially in a highly coupled and non-linear environment such as in the ocean.

There are several difficulties in designing a controller that provides good performance and stability to AUVs. Firstly, in robust control, there is a trade off between stability and performance of a system. This is discussed later in section 3.1. Secondly, the nature of the ocean environment where it is very uncertain. Lastly, only some of the nonlinear vehicle dynamics are known and thus, precise information is needed. Therefore, to cater for such uncertainties, a lot of consideration has to be taken during designing an AUV controller. Other examples of uncertainties are unmodelled dynamics and modelling errors.

Conventional controllers, for example PID, are inadequate for such tasks as they only provide small range of robustness. In addition, the complete dynamics of the vehicle should be known precisely. Therefore, this results in expensive and intensive testing of the vehicle. Hence, a robust control law is introduced to overcome the problem.

Robustness is important for AUVs because the vehicle often has to skim the seabed and to avoid possible obstructions in the uncertain ocean. In addition, the controller has to take into consideration the nonlinear dynamics of the vehicle. This has lead to the objective of the project where a robust controller is to be designed for an underwater vehicle. Apart from reducing the chance of losing an AUV, a robust controller can help to achieve one's mission to explore the underwater world in a safer and more reliable vehicle.

Below are several controllers available that may be able to perform the tasks of AUVs in the uncertain surroundings. The control methods are Proportional, Integral and Derivative (PID), sliding mode, fuzzy logic, neural network, adaptive methods, H-infinity.

1.3 Literature review

Many control methods have been used in underwater vehicles to obtain sufficient robustness and performance in their respective tasks and applications. Examples of these control methods are PID, sliding mode, fuzzy logic, neural network and H-infinity control laws. Each approach has its own advantages and disadvantages during the design stage.

The ***classically designed PID control*** has been widely used due to its simplicity. It produces reasonable robustness and performance. However, this is only true without the presence of uncertainty such as disturbances and sensor noise. Lea [1] observed a slow response and an oscillatory motion in the experimental speed and depth responses, which are subjected to disturbances and noise. Generally, all three subsystems: speed, heading and depth are sensitive to sensor noise. The experiment was carried out using an underwater vehicle model, *Subzero II* designed by students at Southampton University. A brief description of *Subzero II* is presented in section 2.1.

An AUV has been developed by the Norwegian Defence Research Establishment (NDRE) [5]. It was used as a testing vehicle for the propulsion system which uses seawater batteries. The 4.315 m long and 0.735 m in diameter vehicle, has a nominal cruising speed of 2.1 ms^{-1} and displacement of 1.021 m^3. The control systems were uncoupled into steering, diving and speed controls. Basic PI and PD controllers were used in the *NDRE-AUV* subsystems. The vehicle was involved in several extensive open sea tests. However, some steady state errors were discovered in the heading and depth responses, due to the oscillatory motion of the rudder and sternplane, respectively. This is because both rudder and sternplane servos have a relay nonlinearity behaviour which may cause cycle limit (chattering) behaviour if integral

action is added. In general, however, the results of PID control were satisfactory. It was not used because no uncertainty was taken into consideration during the designing stage of the PID controller.

Kojima *et al* [14] had developed an AUV, *Aqua Explorer 2* for inspection of underwater cables. The PID and fuzzy logic controls were applied to the vertical and horizontal motions but only responses from PID are presented in the paper. The vehicle is 3, 1.3 and 0.9 meters in length, height and width, respectively. The maximum depth is 500 meters to reduce the power consumption by the acoustic links and by acoustic transponders. Some experimental work was carried out in a long tank with dimensions of 200, 15, 6 meters for length, width and depth respectively. For heading, the vehicle was required to track a straight cable which was placed at the bottom of the tank. During the test, the vehicle was commanded to leave the cable for 10 seconds. There was an overshoot of 12 degrees when the vehicle resumed to track the cable. It reached again the desired heading after about 20 seconds. There were some fluctuations which were due to the magnetic noise from the electric motor.

An AUV *PURL II* was used for the study of small lakes. It was developed by Laval *et al* and used a PID controller for the heading and propulsion control [15]. The *PURL II* vehicle is small in size to reduce the power consumption and to increase the applications of small AUVs in lakes. *PURL II* carried a wide variety of oceanographic sensors to obtain pictures of the temperature structure within the thermocline before and after a wind event. It used lead acid batteries because they are cheap, easy to maintain and can be recycled. The vehicle weighs about 70 kg in air and has three thrusters for propulsion. For navigation, a depth sensor which is based on pressure, an acoustics altimeter and a flux-gate compass are used. The vehicle can travel as deep as 70 meters and the cruising speed is limited to 50 cms^{-1} in the horizontal and 10 cms^{-1} in the vertical. PROTEUS is used as *PURL II*'s control software and is implemented in C+ +. This real-time scheduler is developed by International Submarine Engineering (ISE). During the experiment, the vehicle was required to follow a heading angle of 340 degrees. It was observed that there were oscillatory motions and that the heading varied between 340 to 348 degrees. In addition, the heading control was affected by noise.

Autosub-1, developed by the Ocean Technology Division (OTD) at the Southampton Oceanographic Centre (SOC) [2], [3], also uses the classical PID controller. The *Autosub* project was funded by the Natural Environment Research Council (NERC) and was designed to collect data for the study of the oceans from the physical, biological, chemical and geophysical sense. In the future, it may be used as an 'underwater satellite'. *Autosub-1* has a shape of a torpedo. It is 7m long and 0.9 in diameter. The maximum depth and range is 2500m and 1000km, respectively using rechargeable lead–acid or manganese alkaline primary batteries. It can travel at speeds of 1.6 ms^{-1} -1.8 ms^{-1}. A Global Positioning System (GPS) antenna is used at the surface for navigation whereas estimation (dead-reckoning) is used when it is submerged. For simplicity, the horizontal and the vertical plane controls of the vehicle are separated into three subsystems. It was found that the experimental responses produced adequate performances.

To increase robustness of a controller, another control method known as the ***sliding mode*** was introduced. It uses the state space model form and an exact description of the system dynamics is required. The method was tested by Lea [1]. It was found that the sliding mode controller produced better performance in the speed control compared to PID, as it is robust to changes in target speed. In addition, the controller is less sensitive to noise. The heading control was noise sensitive because the use of Kalman filtering has produced a side effect where the noise presented in the synthesised yaw rate data was also reduced. The Kalman filter is used to estimate unmeasurable state variables such as the sway and heave velocities of the vehicle. Seube [8] also tested the approach on an underwater vehicle simulation program. This has resulted in accurate tracking only up to 20% of the dynamics uncertainty present. However, in some situations, the sliding mode control method leads to saturation of thrusters.

Another control law that is available is the ***fuzzy logic approach***. However, it was found that it is noise sensitive in all three controls: speed, heading and depth [1]. Thus, fuzzy logic controllers appear to be less robust compared to PID and sliding mode controllers.

A ***neural network-based*** controller can 'learn' to control a process by using the input and output data. It does not require a formal model and it can deal with non-linear

systems. However, the tuning process is very time consuming. According to Seube [8], it took 1000 times longer to train the controller (off-line by simulation) than a direct adaptive controller. Venugopal *et al* [23] implemented an on-line learning method to control the Florida Atlantic University's (FAU) *Ocean Voyager* vehicle. However, the responses produced were very slow during pitching and yawing. Low frequency oscillations were also observed in the yaw response.

There is another intelligent control method, ***reinforcement learning*** which has been tested by the Australian National University on the *Kambara* AUV [20]. The vehicle was developed for exploration and inspection purposes. *Kambara* has an open frame structure and supports five thrusters and two watertight enclosures. The thrusters enable *Kambara* to roll, pitch, yaw, heave and surge. The upper enclosure consists of sensors, computer and an electronics package whereas the lower enclosure consists of batteries and other sensors. Reinforcement learning requires no vehicle model and it generates continuous outputs based on continuous state information by an unusual interpolator. The controller learns in response to a scalar 'reward' signal and the aim is to maximise the total reward over time. As for *Kambara*, the vehicle learns to control its thrusters in response to command and sensor inputs. Currently, the vehicle has only been tested in simulation and no robustness tests have been carried out. The results showed that this method was able to guide the autonomous vehicle to its target but further experiments are needed to verify the good performance from a neural network structure and learning parameters in the real vehicle.

An ***adaptive control method*** was tested on an AUV, (*Omni-Directional Intelligent Navigator*, *ODIN*) [21]. This vehicle was built and designed at the Autonomous Systems Laboratory (ASL) of the University of Hawaii. The vehicle has a near-spherical shape with vertical diameter of 0.61 meter and horizontal diameter of 0.63 meter. It weighs about 125 kg in air and is made of anodizied aluminium. *ODIN* uses Lead Gel batteries to power the thrusters and the CPU, and provides two hours of autonomous operation. The actuator system consists of eight marine propellers and are actuated by brushless motors. For the sensory system, a pressure sensor, sonars and an inertial system are used for depth, position reconstruction and navigation, and attitude and velocity measurements, respectively. Experiments to track a desired trajectory with a trapezoidal velocity profile were carried out. The control law employed is based

upon a proportional-derivative (PD) technique with an adaptive compensation of the dynamics. The basic control law is similar to the classical adaptive controllers. The objective was to have zero steady state error with the presence of constant external disturbances as well as with partial knowledge of the dynamics. An integrator (I-term) is not present in the PD technique because the adaptive actions give advantage over the simple integral actions on the error variables. From the experimental results, the tracking performance of the horizontal plane was satisfactory but was affected by noise present in the sensor. For depth control, the first 50 seconds of depth response was slow. This is because all the dynamic parameters were not known and as a result, the adaptation action took time to respond. The depth response was also affected by the noise present in the pressure sensor. Therefore, the proposed controller was not very robust to the noise present in the sensors.

H-infinity control concentrates upon the frequency domain and it has been around for the past 20 years. This approach involves minimising the maximum value of the closed loop error to disturbances or other parameters over a range of frequencies. A discussion of H-infinity control can be found in section 3.3. The standard H-infinity approach was employed by Silvestre C and Pascoal [7] in their prototype AUV called *MARIUS*. Some sea trials were carried out. The heading response was found to be good regardless of the slow time response during 10 degrees turning. However, for depth control the vehicle was found to be sensitive to wave disturbances close to the surface. This is because the wave disturbances were not taken into account during the depth control computation. Further work to include uncertainty modelling is being carried out by the authors [7].

H-infinity is widely used in other types of vehicles. Hyde *et al* [22], has applied this method to an Advanced Short Take-Off and Vertical Landing (ASTOVL) aircraft. To obtain good performance and robustness over the whole flight envelope, several H-infinity controllers (as plant observers) were gain-scheduled to take into account different airspeed ranges. After a few flight tests, it was found that the H-infinity control provided the required performance and robustness to the aircraft. Consideration of the plant uncertainty and the shorter iteration process during the controller design by this control law are the advantages of H-infinity over the classical control methods in practice.

The H-infinity method was also employed in depth control of a submarine by Liceaga-Castro and Molen [4]. The depth performance was compared with a classically designed PID controller, and the two simulated responses were found to be similar. However, H-infinity gives more stability and performance robustness to any disturbances that occur. Comparisons have also been made between H-infinity and PID controllers for control of modern warships [10]. It was found that the classical PID gave better reduction on the roll and yaw motions but becomes unstable with the presence of uncertainties. In comparison, H-infinity proved to provide better stability and performance to the warship. Another comparison was carried out between the H-infinity/ μ-synthesis approach and the sliding mode control on an AUV developed by the Draper Laboratory / Massachusetts Institute of Technology (MIT) Sea Grant *Sea Squirt* [9]. The H-infinity/ μ-synthesis approach uses the 'loop shaping' weighting functions where the performance transfer function is 'loop shaped' using the frequency domain weighting functions [9]. Tests were simulated using an AUV full nonlinear model. It was found that the H-infinity/μ synthesis approach was better in heading control but was weak in dealing with nonlinearities such as the hydrodynamic drag. Whereas the sliding mode controller performed better in depth control, it was weak in dealing with low frequency unmodelled dynamics uncertainty.

It has been shown that the H-infinity method can produce controllers which are robust to uncertainty. A summary of the research on underwater vehicle control is tabulated for convenience in Table 1.4.

Paper	Group (Vehicle)	Year	Controller	Type	Control
Healey and Lienard [6]	NPS	1993	Heading and depth	Simulation	Sliding mode
Jalving and Storkersen [5]	NDRE (*NDRE-AUV*)	1994	Speed, heading and depth	Experiment	PID
Logan [9]	Draper Laboratory /MIT Sea Grant (*Sea Squirt*)	1994	Heading and depth	Simulation	H_∞, sliding mode
Jalving and Storkersen [5];	NDRE (*NDRE-AUV*)	1995	Speed, heading and depth	Simulation and Experiment	PID
Liceaga-Castro and Molen [4]	Strathclyde (Submarine)	1995	Depth	Simulation	H_∞
Cowling [11]	N/A	1996	N/A	Simulation	H_∞
McPhail and Pebody [2]	Southampton Oceanography Centre (*Autosub-1*)	1997	Speed, heading and depth	Simulation and experiment	PID
Kojima *et al* [14]	University of Tokyo (*Aqua Explorer 2*)	1997	Heading and depth	Experimental	PID and fuzzy logic
Silvestre and Pascoal [7]	Portugal (*MARIUS AUV*)	1997	Heading and depth	Simulation and experiment	H_∞
Lea [1]	Southampton (*Subzero II*)	1998	Speed, heading and depth	Simulation and experiment	PID, sliding mode and fuzzy logic

Gaskett *et al* [20]	Australian National University (*Kambara*)	1998	Heading	Simulation	Neural Network Reinforcement learning
Laval *et al* [15]	Underwater Research Lab., British Columbia, Canada (*PURL II*)	2000	Heading	Experiment	PID
Antonelli *et al* [21]	University of Hawaii (*ODIN AUV*)	2001	Speed, heading and depth	Experiment	Adaptive control

Table 1. 4: Summary of research on underwater vehicle control

N/A = Information is not available, H_∞ = H-infinity

The PID controller is still in use because of its simplicity in design. However, it is noise sensitive and difficult to tune in order to obtain both good performance and robustness. Even though it can be robust, the performance can become unstable in the presence of uncertainties. The design of a sliding mode controller is complex because it requires a complete system model during design. In addition, the quality of the model will deteriorate if the dynamics change. Even though fuzzy logic does not require any modelling in design, extensive tuning is needed in simulation. Therefore, it involves large amounts of computation and is thus time consuming. Like fuzzy logic, neural network control requires long computational times for training. However, the computational time can be reduced with *a priori* knowledge of the underwater vehicle dynamics. Reinforcement learning is still at an early stage whereas the adaptive method is complicated and difficult to understand and implement. As for H-infinity control, the consideration of uncertainty in plant and controller has proved to provide adequate stability and performance of the system.

1.4 Aim of research

The H-infinity method was chosen as the control technique to provide robustness for an underwater vehicle. This is because it has proven good performance and stability in an uncertain situation. Thus, the project is to design a robust controller for an underwater vehicle using the H-infinity technique.

At the moment, only the heading subsystem is tested. A simulation package, *AutoROV* is used to estimate heading responses. This simulator was developed by Lea [1]. A different non linear underwater vehicle simulation program, designed by Feng and Allen [24] was used to simulate the H-infinity controller. The nonlinear simulation program was used to observe how the underwater vehicle behaves with non linear underwater dynamics. In addition, two underwater simulation programs were employed to ensure that the designed H-infinity controller was working well. Several comparisons of heading responses were made with the classical PID method which was designed by Lea [1].

1.5 Layout of the report

The underwater vehicle model, *Subzero II* used in simulation tests is introduced, and the dynamics of an underwater vehicle are discussed in chapter 2. In chapter 3, the H-infinity control technique is described and the design of the H-infinity controllers is briefly discussed. In chapter 4, the underwater vehicle simulation program, *AutoROV* used to obtain the heading responses is presented. The simulated heading results from the H-infinity controller together with the heading response from the PID control are presented in chapter 5. The conclusion of this project and some recommendations for further work are discussed in chapter 6.

Chapter 2

Underwater Vehicle Dynamics

This section briefly discusses *Subzero II*, a small underwater vehicle, and then concentrate upon the dynamics of 'flight' vehicles. The *Subzero II* is described in detail in Lea [1].

2.1 *Subzero II*

Subzero II is a tethered, torpedo-shaped underwater vehicle and is shown in figure 2.1. The hull, which is 1m long and 10cm diameter, is made from Perspex. The nose and tail sections are removable. The propeller is controlled by a samarium-cobalt DC motor. A pack of Ni-Cad batteries is used to supply the power to the motor. It is geared down by the ratio of 5:1. The propeller has a pitch ratio of 1 and blade area ratio (BAR) of 0.12. The control surfaces are the linked rudder and two independent sternplanes, controlling the horizontal plane and vertical plane movements, respectively. They are controlled by model aircraft servos. A very thin fibre optic cable is used as a communication link between the vehicle and the host computer on the surface (PC). The cable is for sending and receiving data, rather than for power supply.

There are two micro controller units (MCU): one on the vehicle and another one inside the PC. The MCU from the vehicle takes data such as sensor speed before sending it to the PC. This measurement is compared with the pilot desired movements before transmitting it back to the vehicle. The pulse width modulator (PWM) is used to adjust the width of pulse of the control signal. There are many sensors installed inside the vehicles, for example, gyroscopes and accelerometers. Currently, the microprocessor is being updated. Figure 2.2 shows the internal layout of *Subzero II*.

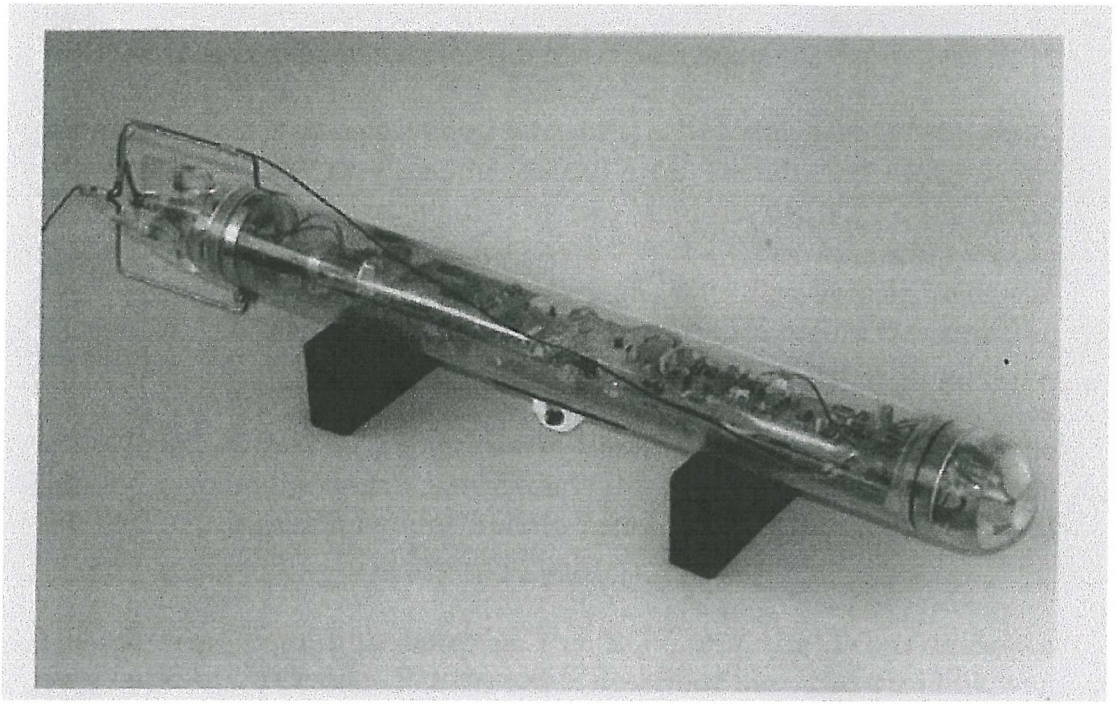


Figure 2.1*: *Subzero II* underwater vehicle

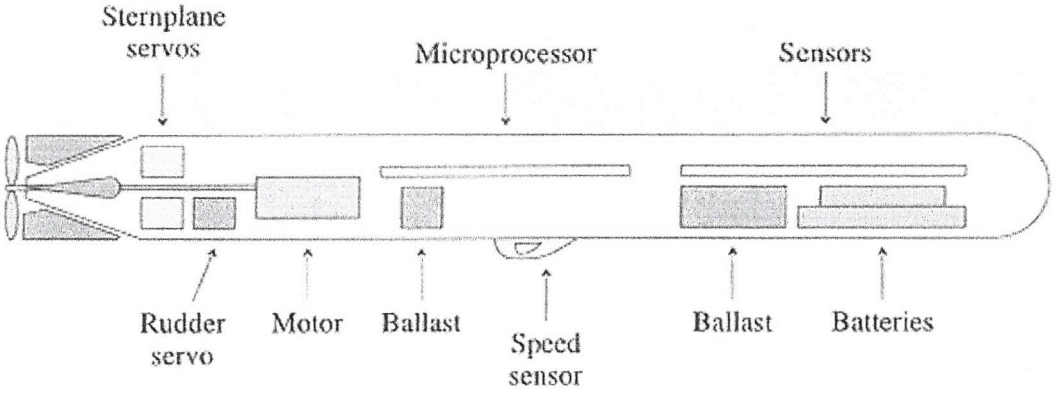


Figure 2.2*: A schematic diagram of the internal layout of *Subzero II* underwater vehicle

* **Figure 2.1 and figure 2.2 are taken from Lea [1]**

2.2 Dynamics

2.2.1 Coordinate systems

The dynamics of an underwater vehicle model can be described by two coordinate systems. They are the global (*earth-fixed*), $X_g\ Y_g\ Z_g$ and local (*body-fixed*), $X_o\ Y_o\ Z_o$ frames. The coordinates and the components are shown in Figure 2.3 below.

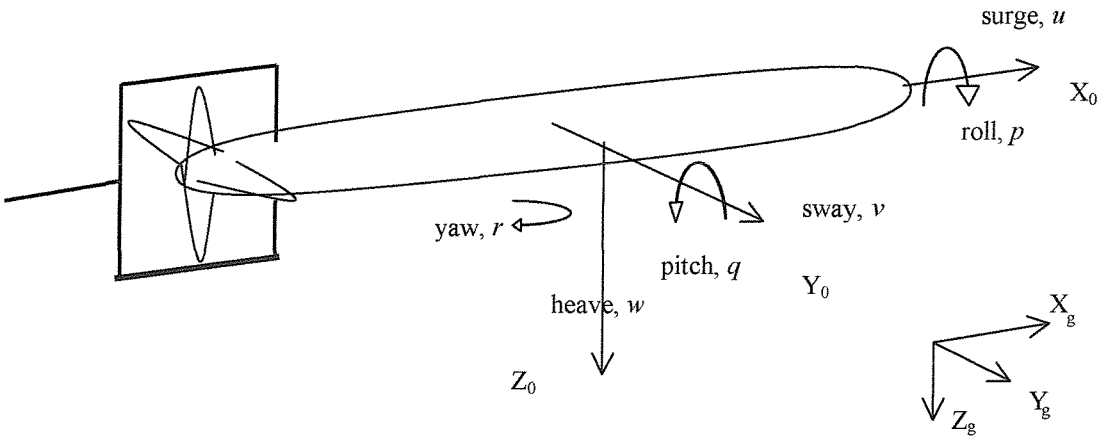


Figure 2.3: A schematic diagram of the *SubzeroII* underwater vehicle coordinate systems and associated variables.

The vehicle's position and orientation vector with coordinates in the global (earth-fixed) frame, $X_g\ Y_g\ Z_g$ is represented as $\eta = [x\ y\ z\ \phi\ \theta\ \psi]^T$. x, y, z are the displacements in the $X_g\ Y_g\ Z_g$ axes, respectively. Where as ϕ, θ, ψ are the rotations around the $X_g\ Y_g\ Z_g$ axes, respectively. ϕ is the roll angle, θ is the pitch angle, ψ is the yaw (or heading) angle, and they are known as the Euler angles. The linear and angular velocity vector with coordinates in the body-fixed frame, $X_0\ Y_0\ Z_0$ is written as $\upsilon = [u\ v\ w\ p\ q\ r]^T$. u, v, w are the translation movements in the $X_0\ Y_0\ Z_0$ axes, respectively. Where as p, q, r are the rotations around the $X_0\ Y_0\ Z_0$ axes, respectively. The relationship between η and υ is related by the transformation matrix, $\mathbf{J}(\eta)$ such that

$$\dot{\eta} = J(\eta)\upsilon$$

$$\text{where } J(\eta) = \begin{bmatrix} c\psi\, c\theta & -s\psi\, c\theta + c\psi\, s\theta\, s\phi & s\psi\, s\phi + c\psi\, c\phi\, s\theta & 0 & 0 & 0 \\ s\psi\, c\theta & c\psi\, c\phi + s\phi\, s\theta\, s\psi & -c\psi\, s\phi + s\theta\, s\psi\ c\phi & 0 & 0 & 0 \\ -s\theta & c\theta\, s\phi & c\theta\, c\phi & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & s\phi\, t\theta & c\phi\, t\theta \\ 0 & 0 & 0 & 0 & c\phi & -s\phi \\ 0 & 0 & 0 & 0 & \frac{s\phi}{c\theta} & \frac{c\phi}{c\theta} \end{bmatrix}$$

and $s\alpha$ = sin α, $c\alpha$ = cos α and $t\alpha$ = tan α. A detailed explanation of the transformations can be found in [1].

2.2.2 Rigid body mechanics

The dynamics of a moving body is based on the Newton second law such that $m \times a = F$ where a is the acceleration produced by the body due to force F exerted on the constant mass m. This is applied for one degree of freedom movement. Force on a body can be applied in different ways and in general the excitation is considered to be in translation or in rotation.

Translational Forces

$$m \times a = F$$

The units are m: kg, a: ms^{-2} and F: $kgms^{-2}$ or N. The translation force F produces a straight and forward acceleration of the mass, m.

Rotational Forces

$$I \times \alpha = T$$

Moment of inertia I (mr^2): kgm^2, angular acceleration α: $rads^{-2}$ and torque T (or moment): $kgm^2(rad)s^{-2}$. The effect of torque is an angular acceleration about an axis of rotation.

2.2.3 Rigid Body (Six Degree of Freedom)

The dynamics of an underwater vehicle can be represented as below [1].

$$M_{RB}\dot{\upsilon} + c_{RB} = \tau \tag{2.1}$$

Subscripts RB represents the rigid body. M_{RB} is the mass matrix and c_{RB} is the vector due to coupling between the motions of the rigid body. The mass matrix, M_{RB} also contain values which are effected by couplings and the matrix is shown below.

$$M_{RB}\dot{v} = \begin{bmatrix} m & 0 & 0 & 0 & mz_G & -my_G \\ 0 & m & 0 & -mz_G & 0 & mx_G \\ 0 & 0 & m & my_G & -mx_G & 0 \\ 0 & -mz_G & my_G & I_x & -I_{xy} & -I_{xz} \\ mz_G & 0 & -mx_G & -I_{yx} & I_y & -I_{yz} \\ -my_G & mx_G & 0 & -I_{zx} & -I_{zy} & I_z \end{bmatrix} \begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \\ \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix}$$

τ is a six-element vector consisting forces $X\ Y\ Z$ in and moments $K\ M\ N$ around the local axes $X_o\ Y_o\ Z_o$, respectively, of an underwater vehicle. The general expressions for translation and rotation forces [26], with couplings are shown below.

Translational

The resultant external force, F can be expressed as

$$F = m\left[\dot{U} + \dot{\Omega}R_c + \Omega(\Omega R_c)\right] = \begin{bmatrix} X & Y & Z \end{bmatrix}^T$$

, where $U = [u\ v\ w]^{\mathrm{T}}$ is the velocity of the origin in axes $X_0\ Y_0\ Z_0$ and $\Omega = [p\ q\ r]^{\mathrm{T}}$ is the angular velocity around the origin in axes $X_0\ Y_0\ Z_0$, as shown in figure 2.3.

$R_c = [x_G\ y_G\ z_G]^T$ is the positions of the center of gravity in the local coordinates $X_0\ Y_0\ Z_0$ system. The second (ΩR_c) and the third $\Omega(\Omega R_c)$ terms are due to coupling effects.

Rotational

For the resultant external moment

$$G = \frac{d}{dt}([I]\Omega) + m(R_c\dot{U}) = [K \quad M \quad N]^T$$

(R_cU) is the coupling effect from the moment (rotation) about an axis and

$[I] = \begin{bmatrix} I_x & -I_{xy} & -I_{xz} \\ -I_{yx} & I_y & -I_{yz} \\ -I_{zx} & -I_{zy} & I_z \end{bmatrix}$ is the moment of inertia around the origin.

The expression of six degrees of freedom (DOF) for individual forces [$X\ Y\ Z$] in and moments [$K\ M\ N$] around the local axes $X_o\ Y_o\ Z_o$, respectively, of an underwater vehicle are shown below.

$$\begin{aligned} m[\dot{u} - vr + wq - x_G(q^2 + r^2) + y_G(pq + \dot{r}) + z_G(pr + \dot{q})] &= X \\ m[\dot{v} - wp + ur - y_G(r^2 + p^2) + z_G(qr + \dot{p}) + x_G(qp + \dot{r})] &= Y \\ m[\dot{w} - uq + vp - z_G(p^2 + q^2) + x_G(rp + \dot{q}) + y_G(rq + \dot{p})] &= Z \end{aligned} \qquad \textbf{(2.2)}$$

$$
\begin{aligned}
&I_x\dot{p}+(I_z-I_y)qr-(\dot{r}+pq)I_{xz}+(r^2-q^2)I_{yz}+(pr-\dot{q})I_{xy}\\
&+m[y_G(\dot{w}-uq+vp)-z_G(\dot{v}-wp+ur)]=K\\
&I_y\dot{q}+(I_x-I_z)rp-(\dot{p}+qr)I_{xy}+(p^2-r^2)I_{zx}+(qp-\dot{r})I_{yz}\\
&+m[z_G(\dot{u}-vr+wq)-x_G(\dot{w}-uq+vp)]=M\\
&I_z\dot{r}+(I_y-I_x)pq-(\dot{q}+rp)I_{yz}+(q^2-p^2)I_{xy}+(rq-\dot{p})I_{zx}\\
&+m[x_G(\dot{v}-wp+ur)-y_G(\dot{u}-vr+wq)]=N
\end{aligned}
\tag{2.3}
$$

m is the mass of the underwater vehicle, x_G, y_G and z_G are the center of gravity of the vehicle in the x, y and z directions, respectively. I is the moment of inertia around the appropriate axes. The highly coupled between the degree of freedom in motions result in nonlinear terms.

The underwater environment is complex for analysis and an AUV can experience many forces in the ocean. There are different types of *forces* produced in the underwater environment. The relationship is described below.

2.2.4 Hydrodynamics forces and moments

The expressions (2.2) and (2.3) above described the motion of an underwater vehicle (kinematics). The relationship of motion and the forces that causes their changing of motion (kinetics) is described below.

The forces produced by a moving underwater vehicle are proportional to position, velocity and acceleration [1], [26]. They are:

1) the *weight* produced by the vehicle.
2) the *buoyancy* or the upthrust of the displaced fluid.
3) the *drag forces* due to velocity or angular acceleration of the vehicle.
4) The *fluid motion* and the added mass due to the velocity and acceleration of the fluid.

5) The *remaining added mass and inertia* which are combined as the dynamics vector.

Forces 1) and 2) are examples of the forces produced when the moving vehicle is proportional to position and this is denoted by $g(\eta)$. Force 3) are proportional to velocity and it is indicated by $D(\upsilon)$. Examples of forces experienced while the vehicle is moving and proportional to acceleration are given in 4) and 5). Added mass is another inertia term and Fossen [27] has defined it as the forces and moments induced by pressure due to a forced harmonic motion of the body. The forces produced by a motion underwater vehicle is also a function of the *actuators* such as the thruster or propeller, rudder and sternplane. This force is denoted with the symbol $B(\upsilon)$.

Therefore, τ can also be expressed as

$$\tau = -M_A\dot{\upsilon} - c_A(\upsilon) - D(\upsilon)\upsilon - g(\eta) + B(\upsilon)u \tag{2.4}$$

where M_A is the matrix of added masses and c_A is the vector of forces and moments due to the coupling effect of the added masses. Subscript A represents the added mass effect. Combining expression (2.1) and (2.4) gives

$$M\dot{\upsilon} + c(\upsilon) + D(\upsilon)\upsilon + g(\eta) = B(\upsilon)u \tag{2.5}$$

Expression (2.5) is the generic equation of motion for an underwater vehicle with $M = M_A + M_{RB}$ and $c = c_A + c_{RB}$. $\boldsymbol{M}$ is the mass matrix and c is the coupling, D is the drag force due to velocity, g is the gravitational force and B is the force from the actuator. υ is the velocity component of the local (body) coordinates. In expression (2.5) above, the weight W and the buoyancy B forces are combined as the gravitational forces g. Interested readers are referred to [26] and [27] for further details of the vehicle dynamics and the evaluation of the equation above.

The generic equation of motion in expression (2.5) can be expanded into six-degree-of-freedom movements. They are the *surge, sway, heave, roll, pitch and yaw*. Surge (u), sway (v) and heave (w) are the velocity movements in the x, y and z local axes, respectively. Whereas, roll rate (p), pitch rate (q) and yaw rate (r) are the rotational (velocity) movements about the x, y and z local axes, respectively. The complete non linear equations of motions can be obtained in Appendix 8.1. The hydrodynamic forces and moments such as added mass and drag are further discussed in [26].

Added mass

This is an additional inertia term which is added, to take into account the mass of surrounding fluid that accelerated with the vehicle. The added mass is defined the proportionality constant which relates the linear and angular accelerations with each of their generated hydrodynamic forces and moments. For example, the inertia force term in the x-axis, X_A is shown as

$$X_A = -X_{\dot{u}}\dot{u}$$

and $X_{\dot{u}}$ is the added mass coefficient. In general, the forces and moments due to added mass can be represented as

$$\begin{bmatrix} F \\ G \end{bmatrix}_A = -\frac{d}{dt}\left([A] \begin{bmatrix} u_r \\ \Omega \end{bmatrix} \right),$$

where $[A]$ is the added mass matrix, u_r is the velocity of the vehicle relative to the sea and Ω is the angular velocity about the origin. For a completely submerged underwater vehicle, like *Subzero II*, the coefficient of the added mass is constant.

Drag

Drag force is due to the square of the *relative motion of the vehicle.* The general expression is

$$Drag = \frac{1}{2}\rho u^2 C_{F0} A_F$$

where ρ is the density of the water (10^3 kgm^{-3}), u is the velocity of the vehicle, C_{F0} is the drag coefficient of the undeflected fin and A_F is the area of the fin. The fin mentioned also applied to the sternplane action. When the fin is deflected, lift is also created. For small angles of deflection, the lift coefficient C_L is proportional to the rudder angle, $C_L = k_1 \delta r$. k_1 is the proportional constant and δr is the deflection of the rudder. Lea [1] assumed that the drag of the vehicle consists of the surge motion and rudder deflection such that

$$Drag = \frac{1}{2}\rho u^2 [C_B A_B + (C_{F0} + k_1{}^2 k_2 \partial r^2) A_F . \tag{2.6}$$

k_2 is another proportional constant, C_{F0} is the drag coefficient when the fin is undeflected, C_B is the drag coefficient for the body, A_F is the area of the fin and A_B is some area related to the vehicle.

The standard total fin drag coefficient is the sum of the undeflected drag coefficient and deflected drag coefficient. In practice, all of the above components in (2.6) cannot be determined separately. Therefore, several 'towing tank' tests were carried out to obtain values for the non-dimensional hydrodynamics coefficient, X_d = (*drag coefficient C* x *area A*). The expression of drag force in term of the coefficients obtained from the towing tank tests are

$$Drag = \frac{1}{2}\rho X'_{uu} u^2 + \frac{1}{2}\rho X'_{\partial r \partial r} u^2 \partial r^2 .$$

(2.7)

X_{uu} and $X_{\delta r \delta r}$ are the non-dimensional hydrodynamics coefficients related to speed and deflection of the rudder, respectively. Comparing expressions (2.6) and (2.7)

$$X'_{uu} = C_{dB} A_B + C_{dF} A_F \text{ and } X'_{\partial r \partial r} = k_1^{\,2} k_2 A_F .$$

where C_{dF} is the total fin drag coefficient.

In the expressions for the equations of motion in Appendix 8.1, the drag force on the x-axis due to the relative velocity in the x-axis is shown as $-X_{uu}|u|u$. Whereas the drag force along the y-axis due to the angular velocity q is expressed as $-M_{qq}\, q|q|$. For further explanation on drag, interested readers are referred to [1].

Weight and buoyancy (gravitational)

In order for the vehicle to be stable, the centre of buoyancy (B) has to be above the centre of gravity (Z). This is shown in Figures 2.4 and 2.5 below. This is related to the *Archimedes' Principle*: when an object is submerged partially or completely in a fluid, it experiences a vertically upward buoyant force, [28]. This force F_B is equal to the weight of fluid displaced by the object. F_B is the buoyant force and it is dependent on the density of the fluid (ρ), gravity (g) and the volume of the submerged object (V). Whereas F_W depends on the mass of the displaced fluid, m and gravity, g. The expressions below are taken from [26].

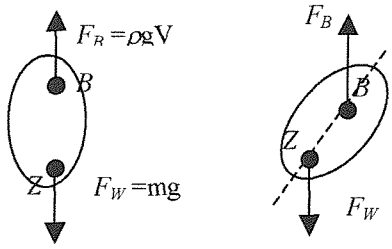


Figure 2.4: Stable condition when B is above G. This condition is called the righting moment.

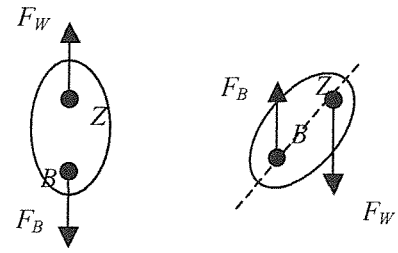


Figure 2.5: Unstable condition when B is below G. This condition is called the overtuning moment.

$$F_W = mg\,[\alpha] \qquad \text{and} \qquad F_B = -\rho g\, V\,[\alpha]$$

where $[\alpha] = [-\sin\theta \quad \cos\theta \sin\phi \quad \cos\theta \cos\phi]^T$ is the transformation from the global to the local frames. For the moments,

$$G_W = R_C F_W \qquad \text{and} \qquad G_B = R_B F_B.$$

where R_C is the position of the center of mass and R_B is the position of the center of buoyancy in the local coordinate system.

Actuators

Generally, AUVs use a propulsion unit and control surfaces to manoeuvre. For *Subzero II*, the propulsion is produced by a propeller. The control surfaces are the rudder which controls the heading (sway); and the sternplane which controls the depth (heave). This is a positive force because it produces the power for the vehicle to move. The surge or forward motion is provided by the thrust force T from the propeller where

$$T = \frac{1}{2}\rho[V_a^{\,2} + (0.7\pi n D_p)^2]\frac{\pi}{4}D_p^{\,2}C_T$$

V_a is the water inflow velocity. For *Subzero II*, V_a is taken to equal to the speed of the vehicle u. C_T is the thrust coefficient which is obtained from tank tests. D_p is the diameter of the propeller, n is the propeller speed and ρ is the density of the water.

The heading subsystem is derived from the simplified version of the sway and yaw equations of motion (8.2) and (8.6), respectively. In sway, the drag force produced by the rudder is $Y_{\delta r}u_o^2\delta r$ and in yaw motion, the resulting moment is $N_{\delta r}u_o^2\delta r$.

In depth control, the heave and the pitch motions are affected by the deflection of the sternplane. For heave, the drag force is $Z_{\delta s}u_o^2\delta s$ and the moment in pitch is $M_{\delta s}u_o^2\delta s$. $Y_{\delta r}$, $N_{\delta r}$, $Z_{\delta s}$ and $M_{\delta s}$ are the non-dimensional hydrodynamic coefficients.

The movements of the underwater vehicle are very dependent upon one another. In addition, the underwater environment is non linear because of the coupling between the degree of freedom. Therefore, the coupling and non- linear behaviour should be taken into account in designing a robust system.

Chapter 3

Theory

In the first section, the robust control theory is discussed briefly. The H-infinity method and in particular the mixed sensitivity H-infinity control employed to obtain a robust system are looked into briefly in the following sections. Several closed loop objectives: disturbance rejection, noise attenuation, optimal control and reference tracking, need to be achieved in order to have a robust system. The design procedure for the H-infinity control is summarised in the last section.

3.1 Robust control

In general, *robust* means *less sensitive* and in control, a robust system is defined as a stable condition regardless of any disturbances that may occur. A closed loop system is more robust with respect to disturbances and noise compared to an open loop system. The analysis of robustness involves a performance criterion called the *sensitivity function.*

Classical robust control uses eigenvectors through eigenvalue problem for analyses where as modern robust control analysis employed the use of singular value. The eigenvector is the corresponding solution to the eigenvalues and eigenvalues exist if there is non trivial solutions to $\boldsymbol{Ax} = \lambda \boldsymbol{x}$, where A is a n x n matrix, λ is the eigenvalue and $\boldsymbol{x} = [x_1\ x_2\ \ldots x_n]^{\mathrm{T}}$. A detail discussion on the eigenvalue problem can be determined in most mathematics textbooks such as [38]. The singular value σ is defined as the positive square roots of the eigenvectors of $Q^{\mathrm{T}}\,Q$, where Q is a complex matrix. Further explanation of singular values can be found in many control textbooks such as [31], [37], [43]. The singular value is preferred above the eigenvectors because the computation of the eigenvectors appears to be numerically sensitive where a small

changes in one matrix element results in large changes in the eigenvectors. In comparison, the standard singular value stability robustness theorem produces a small change in $\sigma(\Delta)$ as a result of a small change in Δ. The history on classical and modern robust control can be found in [43].

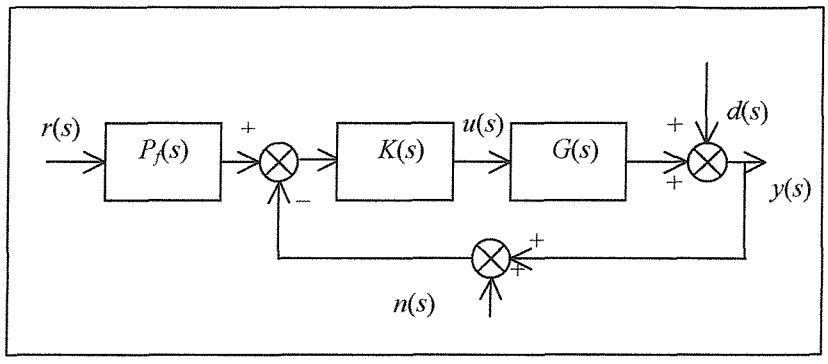


Figure 3.1: Standard SISO feedback configuration

Figure 3.1 shows a diagram of a standard single input-single output (SISO) feedback control system. $P_f(s)$ is a prefilter (not always included) which modifies the setpoint, $r(s)$. $G(s)$ is the plant dynamics and $K(s)$ is a controller. $d(s)$ and $n(s)$ are the disturbances and sensor noise, respectively. $y(s)$ is the output of the system and $u(s)$ is the control signal from the controller. From the figure 3.1 above, two expressions relating the signals can be determined as follows:

$$y(s) = \frac{1}{(1+G(s)K(s))}d(s) + \frac{G(s)K(s)P_f(s)}{(1+G(s)K(s))}r(s) - \frac{G(s)K(s)}{(1+G(s)K(s))}n(s) \tag{3.1}$$

$$u(s) = \frac{K(s)}{1+G(s)K(s)}[r(s) - n(s) - d(s)] \tag{3.2}$$

Expression (3.1) relates the output $y(s)$ with the disturbance $d(s)$, noise $n(s)$ and reference $r(s)$ signals. While expression (3.2) shows the relationship between the control signal from the controller $u(s)$ and the disturbance $d(s)$, noise $n(s)$ and reference $r(s)$ signals.

A sensitivity function is one of the performance criterions used in feedback control systems [29]. It measures the dependency of the overall system's characteristic, $H(s)$ on a particular element in the system. $H(s)$ is the overall closed loop transfer function which is the ratio of the output and the input of the system, such that

$$H(s) = \frac{y(s)}{r(s)}.$$

For example,

$$S_K^H(s) = \frac{\%\Delta H}{\%\Delta K} = \frac{\frac{dH}{H}}{\frac{dK}{K}}. \tag{3.3}$$

where $\%\Delta$ indicates the 'percentage change in'.

The sensitivity is thought to be a better measure of robustness because it tells whether the change in an element of the controller, K will directly affect the overall system characteristic. For example, if $S_K^H(s) = 1$, the change in the controller, K is crucial. This is because an increase in the controller gain will increase the output of the system by the same rate. In comparison, if $S_K^H(s) = 0.2$, the change of the output with respect to the change in the controller is much less. Therefore, it is important to have a small value of $S(s)$ over the frequency range of interest.

The expressions (3.1) and (3.2) can also be written in term of the sensitivity functions. From (3.1), $y(s)$ becomes

$$y(s) = S(s)d(s) + T(s)P_f(s)r(s) - T(s)n(s) \tag{3.4}$$

and from (3.2), $u(s)$ becomes

$$u(s) = R(s)[r(s) - n(s) - d(s)] \tag{3.5}$$

In expression 3.4, $S(s) = \dfrac{1}{1+G(s)K(s)}$ is known as the sensitivity function and $T(s) = \dfrac{G(s)K(s)}{1+G(s)K(s)}$ is the complementary sensitivity function. $S(s)$ relates the output $y(s)$ and the disturbances $d(s)$ where as $T(s)$ relates the noise $n(s)$ and the reference $r(s)$ to the output $y(s)$. $R(s)$ in expression 3.5 is another sensitivity function which relates the control signal, $u(s)$ and the other inputs: $r(s)$, $d(s)$ and $n(s)$. This sensitivity function is sometimes known as the control sensitivity function and $R(s) = \dfrac{K(s)}{1+G(s)K(s)}$.

In [29], it was found that the sensitivity function $S(s) = \dfrac{y(s)}{d(s)}$ in (3.4) is similar to the sensitivity in expression (3.3). Therefore, effect of the controller $K(s)$ on the overall system $H(s)$ can be obtained by making the output $y(s)$ insensitive to disturbances $d(s)$ with a low value of $S(s)$. From expressions 3.4 and 3.5, several closed loop objectives can be determined [31], [37]. They are:

1) Disturbance rejection (robustness with respect to disturbances): $\bar{\sigma}(S(s))$ small
2) Noise attenuation (robustness with respect to noise): $\bar{\sigma}(T(s))$ small
3) Reference tracking (tracking a changing desired output): $\bar{\sigma}(T(s)) \approx \underline{\sigma}(T(s)) \approx 1$.
4) Control energy reduction or optimal control (minimal control input): $\bar{\sigma}(R(s))$ small

(3.6)

σ is the singular value and it is a measure of robustness. The upper, $\bar{\sigma}$ and lower, $\underline{\sigma}$ bar on the singular value indicate maximum and minimum singular values, respectively. The closed loop objectives above can also be approximated in terms of

the open loop transfer function, $G(s)K(s)$ [31], [37]. The approximated requirements are as follow.

1) For disturbance rejection: maximise $\underline{\sigma}(GK)$

2) For noise attenuation: minimise $\overline{\sigma}(GK)$

3) For reference tracking: maximise $\underline{\sigma}(GK)$

4) For control energy reduction or optimal control: minimise $\overline{\sigma}(K)$

(3.7)

It is well known that the disturbances $d(s)$ and the reference $r(s)$ signals are normally concentrated at the low frequency where as the noise $n(s)$ signal dominates at the high frequency region. From expression (3.4) and the closed loop objectives above, it is observed that $S(s)$ relates to $d(s)$ and $T(s)$ relates to $n(s)$ and $r(s)$. Therefore, to reject disturbances $|G(s)K(s)|$ is required to be large ($>>$ 1) at low frequency. This is to ensure that the sensitivity function $S(s)$ is minimized in the low frequency range. On the other hand, noise attenuation is achieved by making $|G(s)K(s)| << 1$ at high frequency so that $T(s)$ is minimised at this range. As long as the system is strictly *proper* at high frequencies (as $|\ G(s)\ |$ will be small) and $K(s)$ does not increase with frequency, $T(s)$ should remain small. A system is said to be *proper* when the number of poles is greater than the zeros. For optimal control of the rudder, it is required that the frequency response of the controller, $K(s)$ to be small in the high frequency range. This is to ensure that $R(s)$ is minimised at high frequency. To fulfil the closed loop objectives (3.7), the frequency response of the open loop transfer function $G(s)K(s)$ is require to have a general shape as in figure 3.2 below.

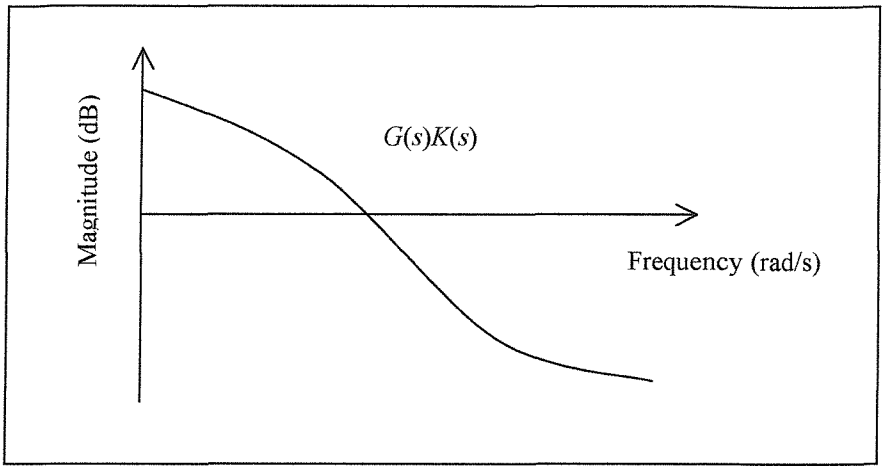


Figure 3.2: An example of an open loop transfer function $G(s)K(s)$

Figures 3.3 and 3.4 show the required frequency response for the sensitivity function, $S(s)$ and the complementary sensitivity function, $T(s)$, respectively.

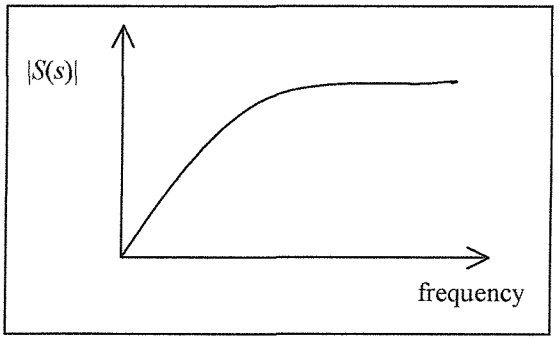


Figure 3.3: Sensitivity function

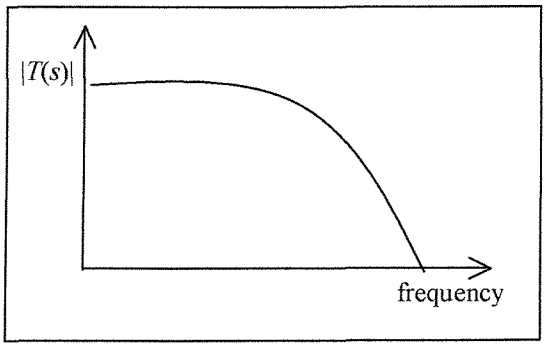


Figure 3.4: Complementary sensitivity function

From figures 3.3 and 3.4, the sensitivity functions can be related such that

$$T(s) + S(s) = 1 \text{ (for SISO)}$$

$$T(s) + S(s) = I \text{ (for MIMO).} \qquad \textbf{(3.8)}$$

SISO is a single input – single output system whereas MIMO is a multiple input – multiple output system. I is an identity matrix. In SISO systems, the singular value can be ignored and the singular value can be taken as the peak magnitude of the Bode plot. From the control objectives above, it is observed that there are some conflicting requirements [37]. These conflicts apply to both control objectives (3.6) and (3.7) above.

i) 2) conflicts with 1) and 3)
ii) 4) conflicts with 1) and 3)

These conflicts can be overcome by selecting different frequency ranges for maximising and minimising the different singular values or H-infinity norms [37]. For example, for disturbance rejection the minimum singular value of $G(s)K(s)$ is required to be large or is maximised at the low frequency range. Where as, the maximum singular value of $G(s)K(s)$ is required to be small or is minimised at the high frequency range. Several control techniques are employed to achieve these closed loop objectives as well as the trade-off in (3.8). The H-infinity method is chosen because it guarantees stability and performance robustness even though the performance might be the same as the classical PID technique, under given conditions. A brief introduction to H-infinity (H_∞) is discussed in the next section. An approach known as the mixed sensitivity H-infinity control, which is designed to take into account the conflicts above is presented in section 3.3.1.

3.2 H-infinity (H_∞)

H-infinity exists in so-called 'Hardy space' which consists of a set of complex-valued transfer functions. Transfer function $F(s)$ has to be analytic (can be differentiated) and in the form of a rational function. Different authors give different interpretations for the Hardy space [30], [31], [32], [33]. The transfer function is bounded by a real constant number, say b; if and only if it is proper and stable [31], [32]. Thus,

$| F(s) | \leq b.$

The H-infinity technique involves minimising the infinity norm of a transfer function, $F(s)$ which is denoted as min $\| F(s) \|_\infty$. The infinity norm of a transfer function can be described as the supremum ('sup') of the magnitude of a transfer function, $F(s)$ where $\| F(s) \|_\infty = \sup | F(s) |$.

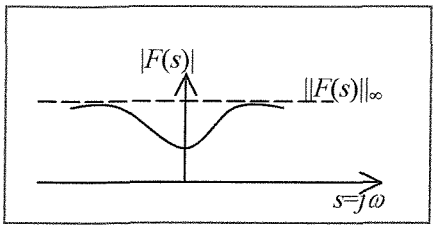


Figure 3.5: Infinity norm

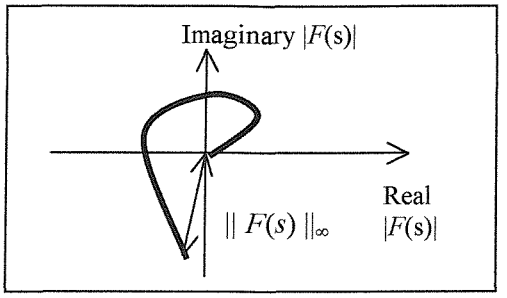


Figure 3.6: Nyquist's plot of $|F(s)|$

Supremum means the *least bound value* of the Hardy space. The H-infinity norm is also defined as a peak gain of a frequency response and the H-infinity norm can be illustrated as in Figure 3.5. A maximum value indicated by 'max' is used in [31] but 'sup' is preferable. This is because the definition of supremum or 'sup' described the infinity norm better in comparison to the definition of maximum or' max'. From a Nyquist's plot of view, $\| F(s) \|_\infty$ is the distance between the origin and the furthest point of the Nyquist's plot [33], which is the maximum $|F(s)|$. This is described in Figure 3.6.

H_2 is another technique that exists in the so-called 'Hardy space'. It takes an average measurement such as the average error. However, the minimisation of the average error or other frequency dependent functions is not as good a design criterion, unlike H-infinity where only the maximum magnitude is considered. H-infinity is a more useful tool as the peak and the frequency at which it occurs, act as good indicators of the response of a system. The definition of the norm H-infinity is summarised as the maximum or peak value of a magnitude of a transfer function as a function of frequency.

3.3 H-infinity control design

The H-infinity control law is discussed in depth in [31] and [33]. It is also briefly described in other references [29], [30] and [43]. A tutorial to H-infinity control is also given in [34] and [35].

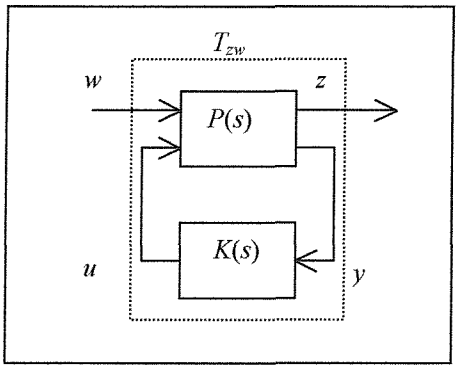


Figure 3.7: General control configuration

Figure 3.7 shows the general control configuration [31] and it is described by

$$\begin{bmatrix} z \\ y \end{bmatrix} = P(s) \begin{bmatrix} w \\ u \end{bmatrix} \tag{3.9}$$

$$\begin{bmatrix} z \\ y \end{bmatrix} = \begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{bmatrix} \begin{bmatrix} w \\ u \end{bmatrix} \tag{3.10}$$

$$u = K(s)\, y \tag{3.11}$$

In state space realisation

$$P(s) = \begin{bmatrix} A & B_1 & B_2 \\ C_1 & D_{11} & D_{12} \\ C_2 & D_{21} & D_{22} \end{bmatrix}$$

The configuration above can be applied to both SISO and MIMO systems. P(s) is the augmented plant consisting the plant dynamics, G(s) and the weighting functions, W(s). A detail diagram showing the block diagram of weighting functions is shown in figure 5.1. A weighting function is a function which is dependent on frequency. K(s) is the controller which helps to stabilise the closed loop system. w is an external input, for example disturbances and noise, and z represents an output or a control error. u is the control input whereas y is the output measurement variable for the feedback. The signals u, y, w and z are to be minimised in order to meet the control objectives. The closed loop from w to z is given by the linear fractional transformation (LFT) as $z = F_l(P, K)\, w$, where $F_l(P, K) = P_{11} + P_{12} K (I - P_{22}K)^{-1} P_{21}$. $F_l(P, K)$ indicates the lower LFT of P with K as the parameter and it is the transfer function T_{zw} from wrapping feedback K around the lower part of P [31], as shown in figure 3.7.

The H-infinity optimal control problem involves finding all stabilising controllers, $K(s)$ which minimise the H-infinity norm of $F_l(P, K)$, $\| F_l(P, K) \|_\infty$. This can be done by minimising the peak of the singular value of $F_l(P, K)$. As mentioned above, $T_{zw} = F_l(P, K)$ where T_{zw} is the closed loop transfer function between the input, w and output, z. In practice, a sub-optimal controller is often simpler to compute and an optimal controller is usually not necessary. The H-infinity sub-optimal control problem is then to find all stabilising controllers K(s) such that $\| F_l(P, K) \|_\infty < \gamma$ where $\gamma > \gamma_{\min}$ and is a positive number. $\gamma_{\min}$ is the minimum value of $\| T_{zw} \|_\infty$ over all stabilising controllers, K(s). The controller K(s) is computed using the solution of two algebraic Riccati equations together with some assumptions obtained in [31]. A brief explanation on the Riccati Equation solution is presented in Appendix 8.2 and [31]. For a SISO system, $\| T_{zw} \|_\infty$ can be represented as the peak magnification of Bode Diagram of T_{zw}. Thus, a stabilising controller can be obtained by attenuation of the peak of T_{zw} below minimum value of gamma, $\gamma_{\min}$.

3.3.1 Mixed sensitivity H-infinity control

Mixed sensitivity H-infinity control is an approach available to obtain an H-infinity controller. As the name suggests, more than one sensitivity functions is involved. The method involves shaping the sensitivity functions as desired to determine the desired specifications design. This is useful when shaping of the sensitivity functions over different frequency range is required, such as in the closed loop objectives in section 3.1 above. The desired specifications are combined into a single infinity norm of the form $\| N \|_\infty \leq 1$ [43].

From [31], $\|N\|_\infty = \max_\omega \bar{\sigma}(N(j\omega)) < 1$ and N can be assigned as

$$\|N\|_\infty = \left\| \begin{matrix} S(s)W_1(s) \\ R(s)W_2(s) \\ T(s)W_3(s) \end{matrix} \right\|_\infty \tag{3.12}$$

where $W_1(s), W_2(s)$ and $W_3(s)$ are the respective weighting functions.

To shape the sensitivity functions, an upper bound is required. This boundary is known as the weighting function, $W(s)$, as presented above. For example, to have a low sensitivity function $S(s)$ at the low frequency range, as in figure 3.2 , the weighting function is recommended to have a shape of a low pass filter. This is because

$$\| S(s)W_1(s) \|_\infty \leq 1$$

$$\Rightarrow \max_\omega \bar{\sigma}(S(s)W_1(s)) < 1 \tag{3.13}$$

From the property of the singular value in [43], $\bar{\sigma}(AB) = \bar{\sigma}(A)\bar{\sigma}(B)$. Thus from (3.13) above,

$$\Rightarrow \overline{\sigma}(S(s))\overline{\sigma}(W_1(s)) < 1$$
$$\Rightarrow \overline{\sigma}(S(s)) < \frac{1}{\overline{\sigma}(W_1(s)}.$$

(3.14)

Further explanation on mixed sensitivity H-infinity control can be obtained in [31], [37] and [43].

3.3.2 Selection of the weighting functions

A weighting function is a function which is dependent on frequency. As stated above, the inverse of a weighting function, $W(s)$ acts as an upper bound for its respective sensitivity function, as shown in expression (3.14). The weighting function can also help with the design of an H-infinity controller by specifying objectives for a design. Examples of common weighting functions are the performance weighting function $W_1(s)$ and robustness weighting function $W_3(s)$. The performance weighting function, $W_1(s)$ is used to help in the attenuation of disturbances, whereas the robustness weighting function, $W_3(s)$ is employed to reject as much noise as possible and to improve tracking performance. From expression (3.12) above, the weighting functions, $W_1(s)$ and $W_3(s)$ should be able to minimise the magnitude sensitivity function, $|S(s)|$ and the complementary sensitivity function, $|T(s)|$ in the low and high frequency ranges, respectively.

No specific formula is available to find a suitable weighting function for a specific design. Therefore, the form of the weighting functions depend mainly on the desired specification of the designers. A weighting function can be as simple as a constant, for example 0.3, and as complicated or complex as a high order polynomial, for example $\frac{2s^2+4s+9}{s^3+4s^2+s+1}$. Some designers consider the selection of the weighting functions as a tuning parameter but some designers use expressions such as in [31]. However, the transfer function of the weightings should be stable and proper. There are also some restrictions in order to achieve a robust system [43].

1) The 0dB crossover frequency of $W_1(s)$ should be smaller than $W_3^{-1}(s)$. This is important for the validation of $|S(s)| < 1/|W_1(s)|$ and $|T(s)| < 1/|W_3(s)|$. In addition, it also help to achieve the robust performance $\||S(s)W_1(s)| + |T(s)W_3(s)|\|_\infty \leq 1$. Figure 3.8 shows the boundaries for performance and robustness.

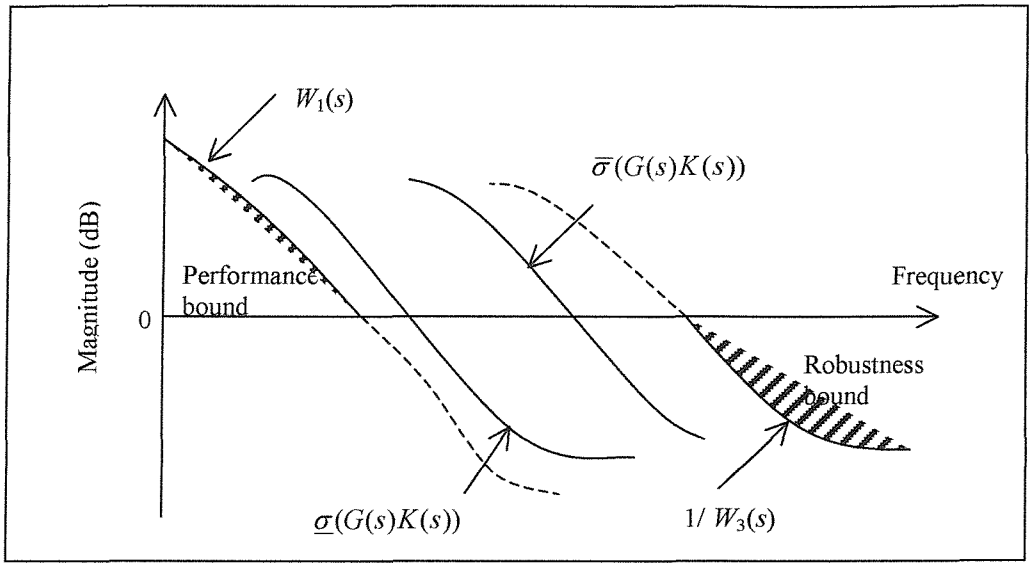


Figure 3.8: Performance and robustness boundaries

2) The roll-off rate of $W_3(s)$ must be at least -20dB/decade to filter out the noise in the high frequency range.

It is advisable to use low order model for the weightings. First order weighting functions are often sufficient to help attaining desired specifications [31].

3.3.3 Bilinear Axis Shifting Transformation

The bilinear axis shifting transformation is a technique used to overcome marginally stable controllers in H-infinity control. An H-infinity controller can be marginally stable when [43]

1) the plant has poles or/ and zeros on the imaginary ($j\omega$) axis

2) the augmented plant (P_{12} or P_{21}) has $j\omega$ - axis zeros (including zeros at infinity) which resulted from a rank deficiency matrix D_{12} or D_{21} in the state space augmented plant, $P(s)$.

The procedure of the transformation is as follow.

1) The plant $G(s)$ is mapped from the s-plane to the $\tilde{s}$ -plane using the formula (3.15) below.

$$s = \frac{\tilde{s} + p_1}{\frac{\tilde{s}}{p_2} + 1} \tag{3.15}$$

where p_1 and p_2 are the endpoints of the diameter of the circles disk region.

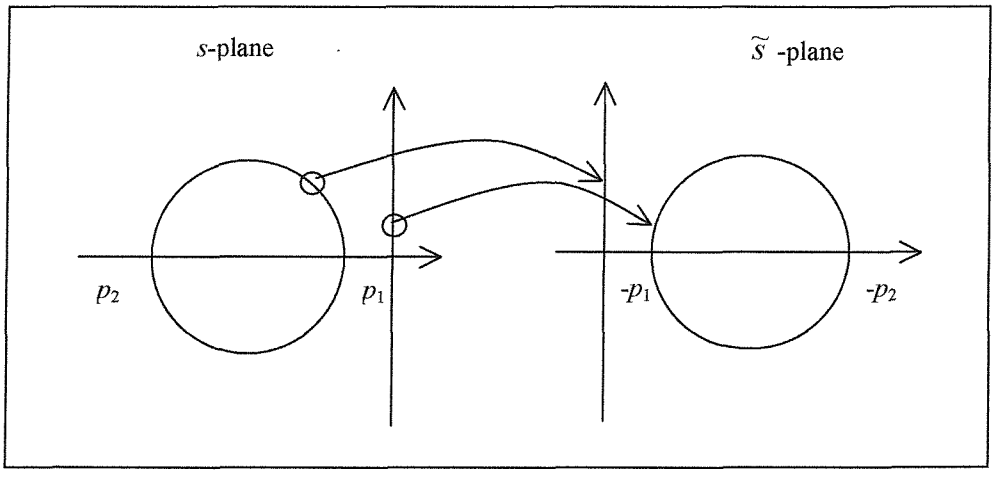


Figure 3.9 Bilinear Axis Shifting Transformation

2) Find the controller $K(s)$ with the new shifted pole or/ and zero in the $\tilde{s}$ -plane. This is similar to solving $\min_{\tilde{K}(\tilde{s})} \left\| \tilde{T}(j\tilde{\omega}) \right\|_\infty \leq 1$.

3) The controller $\tilde{K}(\tilde{s})$ is shifted back using the inverse bilinear transformation and the formula used is

$$\tilde{s} = \frac{-s + p_1}{\frac{s}{p_2} - 1} \tag{3.16}$$

If the design specification is not met, the parameter p_1 is further adjusted. As discovered during several simulations in section 5.5.3, the location of point p_1 affects the size of the deflection for the rudder and thus the heading response of the heading control. Further discussion and few examples of the bilinear axis shifting transformation can be found in [36], [43].

3.4 Summary

A *brief outline* to obtain a stabilising controller, $K(s)$ is as follows:

1) The Bilinear Axis Shifting Transformation is employed if there are any poles or/ and zeros in the plant, $G(s)$ on the $j\omega$-axis.

2) The weighting functions are selected.

3) The plant dynamics and the weighting functions are augmented to form a plant called the augmented plant, $P(s)$ such that

$$\begin{bmatrix} z \\ y \end{bmatrix} = [P] \begin{bmatrix} w \\ u \end{bmatrix}.$$

In addition, $u(s) = K(s)\, y(s)$.

4) The H-infinity optimal controller is computed using, for example, mixed sensitivity H-infinity control.

5) The Inverse Bilinear Transformation is employed if the Bilinear Axis Shifting Transformation was used earlier.

6) The optimal control signal, $u(s)$ can be determined from $u(s) = K(s)\, y(s)$. This is an optional step if the control signal is required.

Chapter 4

AutoROV: Underwater Vehicle Simulation Package

Underwater vehicles are complex robotic systems and usually very expensive. Consequently, it is desirable to test control systems and investigate vehicle performance using simulation, at least initially.

4.1 Background research

The introduction of the David Taylor Naval Ship Research and Development Center (DTNSRDC) has changed the development of computer simulation for underwater vehicles. An early program written by Kapsenberg [39] for a remotely operated vehicle (ROV) was developed in 1985. It was catagorised as 'a low budget (ROV) simulation package'. It used the DTNSRDC's equations and had the same structure as today's simulations. It was written for a microcomputer and acted as a demonstrator.

A different ROV simulation was developed in 1992 by Kalske [40], at the Technical Research Center of Finland. This simulator also uses the DTNSRDC's equations and can simulate either bluff-body or streamlined underwater vehicles. However, these vehicles must use thrusters for manoeuvring. The simulator has been tested against an actual ROV. In most cases, the responses were compatible.

In 1995, Lauvdal *et al* [25] developed a simulation toolbox for 16 different types of marine and flight vehicles including AUVs. The model of the AUV is controlled by rudder and propeller. It is based on the Naval Postgraduate School (NPS) and is

assumed as a block shaped body. The vehicle states which are the output from the simulation test are represented in graphical form. Different types of control law can be included in the simulation program. The program is run on a PC under UNIX with MATLAB. For the underwater vehicle simulation, the control signals: the commands for the motor and deflection of the fins (against time) are needed. These outputs from the controller have to be computed from another file or software before adding it to the program. Thus, the simulation needs to be expanded further if a user wants his or her controller to be simulated within the package.

A simulation package is currently being developed for the Florida Atlantic University (FAU) AUV: *Ocean Explorer* [41]. The research was supported by a 5-year ONR MURI project and is jointly carried out by FAU and NPS. The simulator uses a six-degree of freedom nonlinear AUV model. It forms a closed loop process between the simulator and the AUV controllers, where the simulator generates values to the controllers which in return sends actuator commands back to the simulator. The hardware-in-loop (HIL) is an extension of the package, to study control and visualisation. Initially, the simulation package was implemented on an SGI Irix 5.3. For flexibility, it can be ported to the LINUX operating system. It is mainly used for navigation, validation and integration, and for testing purposes. However, it can only simulate a couple of AUVs: *Ocean Explorer* and the *NPS AUV*. In addition, it requires an advanced operating system such as LINUX and IRIX which makes it a complex and expensive simulator.

A more recent simulator was implemented by NPS [42]. It was tested using the *ARIES* vehicle. The dynamic behaviour is obtained from the MATLAB and SIMULINK software, and for 3D graphical display of a virtual scene, the Virtual Reality Modelling Language (VRML) is used. No high level programming is required by the user. In addition, other new features can be added into the package such as the disturbances.

There are also other simulators such as the EUROSIM, Multi-Vehicle Simulator (which uses a virtual underwater world simulator), AUV SIM (for ROVs and AUVs from *H-Scientific*). Generally, most recent simulation packages require advanced operating systems such as LINUX and UNIX platforms. Thus, high level programming and more expensive software are needed.

AutoROV, developed by Lea [12], is a simple and an user-friendly simulation package. It was developed from FAU's program and modified for the use of *Subzero II* in 1998. The underwater vehicle model, *Subzero II* is designed by the Southampton Group [1]. The torpedo-shaped ROV is 1m long and 10cm in diameter. It is driven by a single propeller and guided by four control surfaces. *AutoROV* is written in C code using Borland C++ (version 4.52) and it runs on a PC. The units have been changed to the standard SI units rather than the American units. Originally the inputs were the motor command and the deflections of the control surfaces (against time) i.e the rudder and sternplanes. They have been modified to commanded speed, heading and depth (against time) in order to allow the addition of new controllers into the package. Other features such as uncertainties can be included and other flight vehicles can also be simulated, given the hydrodynamic coefficients and vehicle dimensions. Tether dynamics, drag and bending characteristics are also available in *AutoROV*. Thus, a flight underwater vehicle with tether can also be tested. In *AutoROV*, there is an option file which allows the user to select the conditions for the flight vehicle and surroundings. For example, time delays of the motor and fins, and sensor noise can be included. The simulation program can still be used by selecting the fixed controller option. The results from the simulation are represented in both tabulated and graphical forms. All of the vehicle states, as well as the response from the Kalman filter and sensors, are presented. The *AutoROV* simulation package is discussed further in Section 4.2.

4.2 The *AutoROV* Simulation Package

The simulation is described and some of the features developed to access different control strategies are presented.

Velocities such as *surge, sway, heave, roll, pitch and yaw* can be found using *AutoROV*. Surge (u), sway (v) and heave (w) are the velocity movements in the x, y and z local axes, respectively. Whereas, roll rate (p), pitch rate (q) and yaw rate (r) are the rotational (velocity) movements about the x, y and z local axes, respectively. The

equations of motions can be obtained in Appendix 8.1. The global frame movements, $\eta = [x\ y\ z\ \phi\ \theta\ \psi]$ are obtained using the transformation matrix **J**. x, y, z are the positional and ϕ, θ, ψ are the rotational movements in the global X_g, Y_g and Z_g axes, respectively.

AutoROV categorises the six-degree-of-freedoms into three subsystems. They are the speed, heading and depth subsystems. The speed subsystem takes into account the *surge* movement as well as the roll motion. The heading considers the sway and the yaw positions. The heave, pitch and the depth are described in the depth subsystem. Some validation tests were carried out and the results, which can be found in [12], were mostly compatible.

Before running the simulation program, all state variables, for example the speed state, are set to their initial values. The hydrodynamics and also the dimension of the vehicle are also assigned for computing matrices **M** and **F**. The velocity components $\upsilon = [u\ v\ w\ p\ q\ r]^T$ are found by either of the two numerical integration methods: Euler or the Improved Euler methods, from

$$M\dot{\upsilon} = F$$
$$\Rightarrow \dot{\upsilon} = M^{-1}F$$

When no method is stated, the Euler method is used. The position and orientation of the vehicle, $\eta = [x\ y\ z\ \phi\ \theta\ \psi]^T$ are found by using the transformation matrix **J**, from the velocity vector components $[u\ v\ w\ p\ q\ r]^T$, where

$$\dot{\eta} = J(\eta)\upsilon$$

The transformation matrix **J** is shown as expression (2.1) in chapter 2. $\dot{\eta}$ is then integrated using either of the numerical integration methods. If the disturbance option is chosen, the Euler method is used. The results are represented in two forms, in tabulated form and / or a graphical form. The tabulated results for both the η and υ components are found in the *rov.out* file. The graphs are displayed using an *Excel Program* under *rov.xls*.

4.2.1 Other Features

As mentioned above, various responses can be obtained, such as the speed (u), heading (ψ) and the depth (z) of the vehicle. Currently, there are four types of controllers included in the package. They are the PID, sliding mode, fuzzy logic and self-tuning regulator controllers. Disturbances and sensor noise can be added to the system to test, for example, the robustness of the controllers. There are other extra features such as the time delays of the motor, rudder and sternplane actuators. In addition, tether dynamics, bending and drag features, are included in *AutoROV*, although a flight vehicle architecture is assumed currently.

AutoROV is flexible and can simulate other flight vehicles. The features are helpful in creating an 'almost' real scenario for operating an underwater vehicle in, for example a surveying application. In *AutoROV*, the subsystems are assumed to be uncoupled where the result of one control axis does not affect the others. Therefore each axis can be investigated separately.

An overview of the package is shown in the flowchart in Figure 4.1. Readers are referred to [12] for further details of the *AutoROV* simulation package and its program listings. Some examples of heading responses obtained from *AutoROV* can be found in section 5.5.

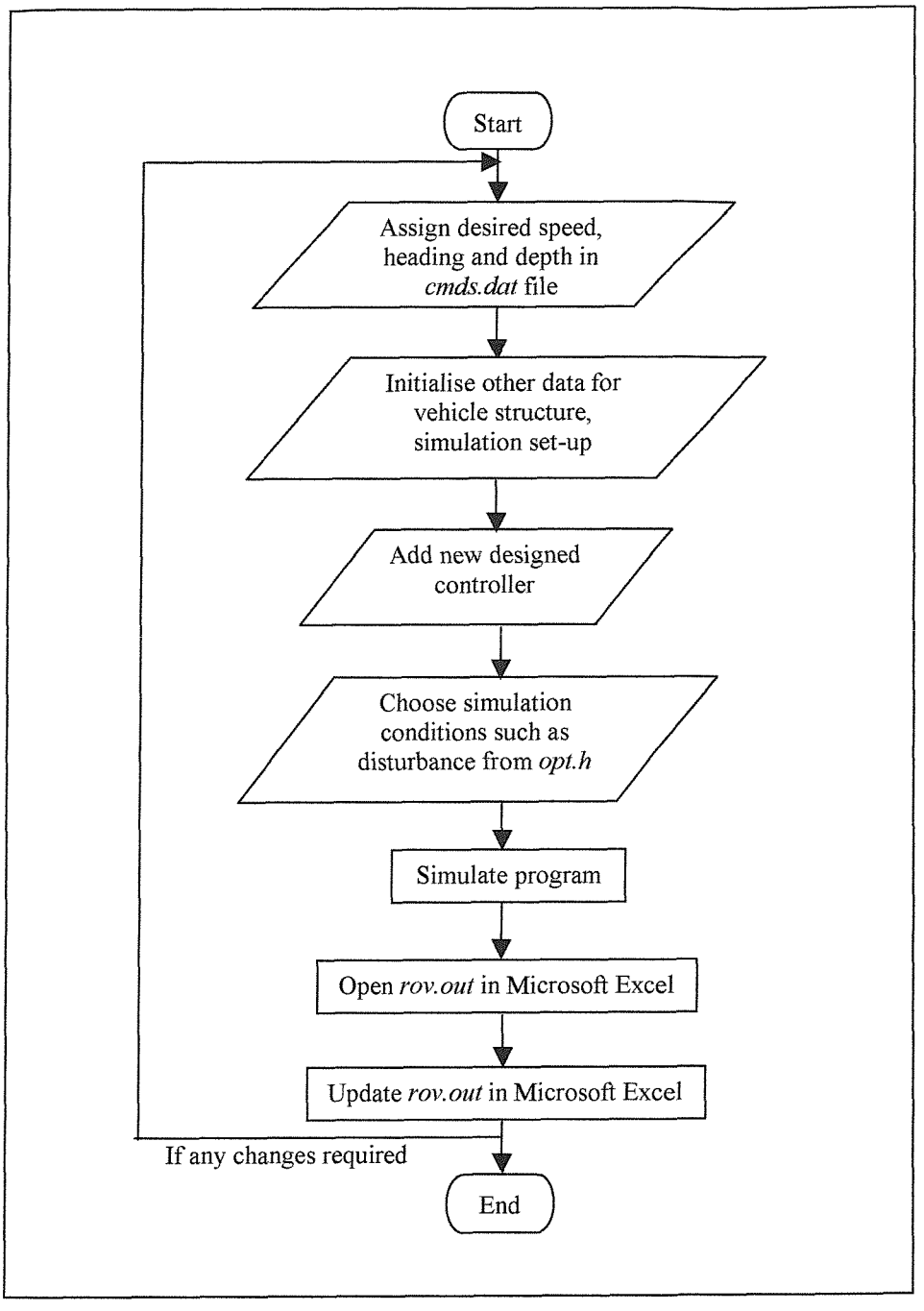


Figure 4. 1: *AutoROV* flow chart

The simulation has already been used to predict the response of *Subzero II* under different control architectures and with a range of input demands. Comparison with experimental trials data has shown that the simulation can indeed describe the vehicle behaviour and it has been shown to provide a valuable testbed for controllers.

4.3 Conclusion

Some initial responses of an expensive system, for example an AUV, are important as it may give some useful information about its behaviour when disturbances are present. This can be achieved by modelling and running some simulation tests. *AutoROV* is a simple and user-friendly underwater vehicle simulation program. Any underwater vehicles can be simulated using *AutoROV*, provided that the hydrodynamics coefficients are known.

Chapter 5

Heading control: Results and Discussion

This section discusses the design of an H-infinity controller for heading control of an underwater vehicle. The heading responses were obtained using the *AutoROV* underwater vehicle simulation package. These results are then compared to the responses obtained from a digital PID controller designed by Lea [1].

Two steps were carried out to obtain a heading response of an H-infinity controller for the underwater vehicle, *Subzero II*.

1) The H-infinity controller was designed using the MATLAB robust control toolbox. Several criteria such as the robust stability and robust performance were analysed. The basic idea of robust synthesis is that a controller is designed based on the frequency domain specification specified by the selected weighting functions $W(s)$.

2) The discretised controller was then included into the simulation package *AutoROV* to obtain the simulated heading responses.

5.1 MATLAB Robust Control Toolbox

The MATLAB robust control toolbox employs the standard H-infinity control technique. This approach is further explained in section 3.3. In the MATLAB robust control toolbox, the transfer function for the plant, $G(s)$ and the chosen weighting functions, $W(s)$ are needed to design an H-infinity controller, $K(s)$ for the heading control. The weighting function, $W(s)$ is a frequency dependent function and it represents the frequency specification for the design of the controller, $K(s)$. The Bilinear axis shifting transformation is employed as the plant in use has a pole on the imaginary axis. This is needed to avoid having a marginally stable, H-infinity controller. Due to the arrangement in the MATLAB robust control toolbox as shown in figure 5.1, the plant $G(s)$ consists of the transfer functions of the linearised heading subsystem and the rudder dynamics. This plant and the selected weighting functions are then augmented as $P(s)$. By solving the Riccati equations, an expression for a stable H-infinity controller is obtained. The algebraic Riccati solution can be found in appendix 8.2. The general arrangement for robust control is shown in Figure 3.9.

Figure 5.1 shows the arrangement used in MATLAB robust control toolbox for computing a H-infinity controller using the mixed sensitivity H-infinity control approach. The remaining variables are described below.

$w(s)$	Input signal	$z_1(s)$	Corrected error signal
$e(s)$	Error signal	$z_2(s)$	Corrected control signal
$u(s)$	Control signal	$z_3(s)$	Corrected output signal
$y(s)$	Output signal		

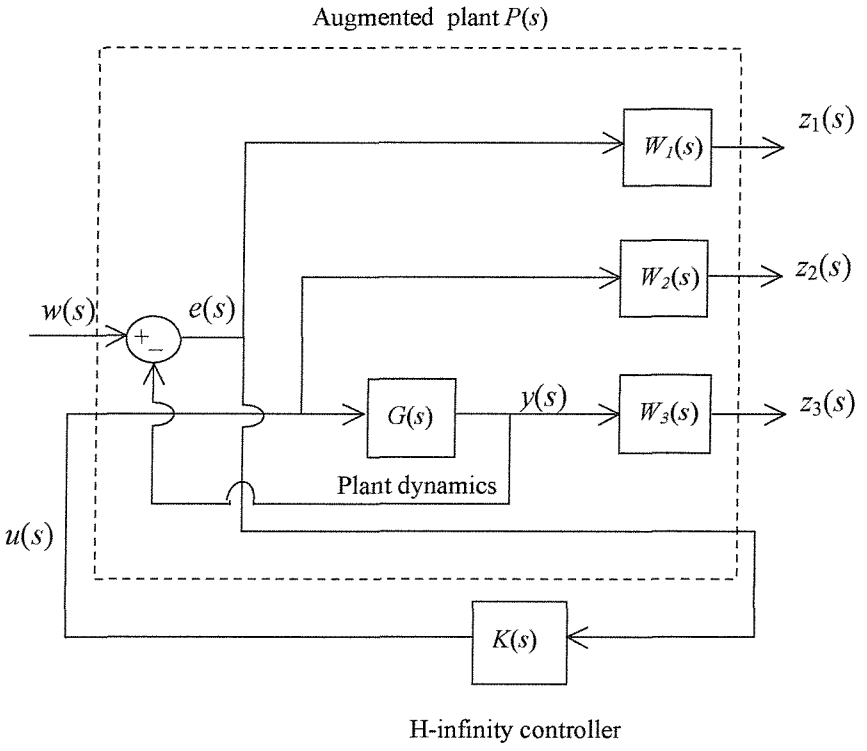


Figure 5.1: Plant Augmentation used in MATLAB Robust Control Toolbox. This diagram is reproduced from the MATLAB Robust Control Toolbox User Guide.

5.2 Control objectives for heading control

A system is said to be robust when it manages to withstand uncertainties such as noise and disturbances in the system. This can be achieved by fulfilling the closed loop objectives listed in section 3.1. In *mixed sensitivity H-infinity control*, the desired performance and robustness objectives are achieved by satisfying the inequalities (5.1) -(5.3) below, where the weighting functions are inversely related to the respective sensitivity functions such as below [30], [34] and [35].

For disturbance rejection:

Sensitivity function, $S(s)$ < Performance weighting function, $W_1(s)^{-1}$ **(5.1)**

For optimal control:

Control sensitivity function, $R(s)$ < Control weighting function, $W_2(s)^{-1}$ **(5.2)**

For noise attenuation:

Complementary sensitivity function, $T(s)$ < Robustness weighting function, $W_3(s)^{-1}$ **(5.3)**

The sensitivity functions were introduced in section 3.1 as a measure of robustness. To achieve the closed loop objectives for the heading control for the underwater vehicle model, *Subzero II*, the sensitivity functions: $S(s)$, $R(s)$ and $T(s)$ are expected to have a certain behaviour. This is discussed in further detail below.

5.2.1 Sensitivity function, $S(s)$

The sensitivity function, $S(s)$ relates the error signal $e(s)$ and the output $y(s)$. In this case, the error signal is the heading error while the output is the actual heading. To reject disturbances, the gain of the sensitivity function, $S(s)$ is required to be as low as possible at low frequencies [31], [37] as described in figure 3.3 in chapter 3. This is done to eliminate the disturbance signal, which by nature, is concentrated at low frequencies. According to Doucy *et al* [42] the wave disturbances lie between 0 and 1.5 rad/s. To improve the sensitivity function, a performance weighting function $W_1(s)$ is added to the design. It is common for $W_1(s)$ to have the shape of a low pass filter for disturbance rejection. This ensures that the inequality (5.1) is fulfilled.

5.2.2 Control sensitivity function, $R(s)$

The control sensitivity function $R(s)$ relates the desired control input $u(s)$ and the output $y(s)$. From expression (3.5), $R(s)$ is related to both disturbances and noise. As mentioned above, the disturbances are concentrated in the low frequency range. However, the noise is concentrated at the higher frequencies [31], [37]. Therefore, the gain of $R(s)$ needs to be low, preferably below 1, in both low and high frequency regions.

5.2.3 Complementary sensitivity function, $T(s)$

The complementary sensitivity function relates the output $y(s)$ and the desired input $w(s)$. Here, the output is the actual heading while the input is the desired heading. To reduce the noise contamination, the roll-off rate for the complementary sensitivity, $T(s)$ must be at least - 20 dB/decade at high frequency. In addition, the frequency response is required to be at least –20dB at 10 rad/s [47]. Expression 3.4 shows that the output signal is maximised by lowering the value of $T(s)$ for noise. Since noise is concentrated at high frequencies, a low gain of $T(s)$ is needed here. To reject the disturbance from the output signal, a function having a high pass filter shape, which is represented by $W_3(s)$, is placed on the output channel.

5.2.4 Other requirements

The above desired closed loop objectives are mainly for robustness. There are other requirements on the performance and robustness of the heading subsystem for an underwater vehicle. There are listed as below.

1) A low steady state error ~ 1%.

2) No, or very low, overshoot to avoid collision with other objects. For example, icebergs during under-ice surveys or hitting the mother ship during docking.

3) Fast heading response with rise time between 5s and 8s. If the heading response is very fast, for example 2s, overshoot is possible. On the other hand, very slow heading response may be time consuming.

4) Reduce the effect of system delay of 0.7s for *Subzero II* as suggested by Lea [1] (within the closed loop 1 Hz bandwidth).

5.3 H-infinity control design

The stages of the H-infinity control design were briefly explained in section 5.1 above. In this section, these stages are discussed further. The stages that are involved in the design of a H-infinity control are, assignment of the plant and weighting functions, the designing the H-infinity controller and the discretisation of the H-infinity controller by the Bilinear transformation method.

5.3.1 Assign transfer functions $G(s)$ and $W(s)$

Plant G(s)

The heading control is composed of three state variables. They are sway speed, $v(t)$, yaw rate, $r(t)$ and heading, $\psi(t)$. The equation of motion for sway and yaw are found in Appendix 8.1. A few assumptions had been made to produce a simplified transfer function for the heading control and the assumptions are as below [1].

a) speed is constant: $u = u_0$
b) second and higher order terms are ignored
c) angle for roll motion is zero, $\phi(t) \approx 0$
d) angle for pitch motion is zero, $\theta(t) \approx 0$ so that the vehicle travels horizontally.

The simplified transfer function for the heading control, without a time delay, is:

$$\frac{\psi}{\delta r} = \frac{-14.1s - 20.95}{s^3 + 10.78s^2 + 15.17s} \tag{5.4}$$

where δr is the actual deflection of the rudder.

In the MATLAB robust control toolbox (figure 5.1), the output and the input of the plant $G(s)$ are y and u, respectively and

$$\frac{y}{u} = \frac{\psi}{\delta r_d}$$

where ψ is the actual heading and δr_d is the desired rudder command. To obtain the ratio of $\frac{y}{u}$ above, the rudder dynamics which will be explained below, need to be included into the plant, $G(s)$. This is because the linearised heading subsystem above (5.4) is in the form of $\frac{\psi}{\delta r}$. Therefore, $\frac{\psi}{\delta r_d}$ was obtained as below.

$$\frac{\psi}{\delta r_d} = \frac{\delta r}{\delta r_d} \times \frac{\psi}{\delta r} \tag{5.5}$$

$$\Rightarrow \quad \frac{\psi}{\delta r_d} = \textit{(Rudder dynamics) x (Linearised heading subsystem)}$$

The rudder dynamics given below (5.6) have a small effect on heading control. This is explained further in Lea [1].

$$\frac{\delta r}{\delta r_d} = \frac{0.9 \times 7.69}{s + 7.69} \tag{5.6}$$

$$\Rightarrow \frac{\delta r}{\delta r_d} = 0.9\left(\frac{1}{\frac{1}{0.13}s + 1}\right)$$

$$\Rightarrow \frac{\delta r}{\delta r_d} = 0.9\left(\frac{1}{\frac{1}{\tau}s + 1}\right)$$

The expression for the rudder dynamics (5.6) above is in a lag form which indicates that the rudder of the underwater vehicle is lagging the rudder command with a time constant, τ of 0.13 s. The final value of the steady state position of the rudder deflection, δr is taken as 0.9 of the demanded rudder command, δr_{d}. This is due to the play and backlash effect in the servo system which can prevent the rudder reaching the exact demanded position.

Hence, using the formula in 5.5, the plant $G(s)$ used for the design of the heading H-infinity controller is

$$G(s) = \frac{0.9 \times 7.69}{s + 7.69} \times \frac{-14s - 20.95}{s^3 + 10.78s^2 + 15.17s}$$

=>

$$G(s) = \frac{-97.59s - 145}{s^4 + 18.47s^3 + 98.07s^2 + 116.7s}$$

with factorisation of the denominator

$$\Rightarrow \qquad G(s) = \frac{-97.59s - 145}{s\left(s^3 + 18.47s^2 + 98.07s + 116.7\right)} \tag{5.7}$$

The arrangement for the heading H-infinity control in MATLAB robust control toolbox is shown as below.

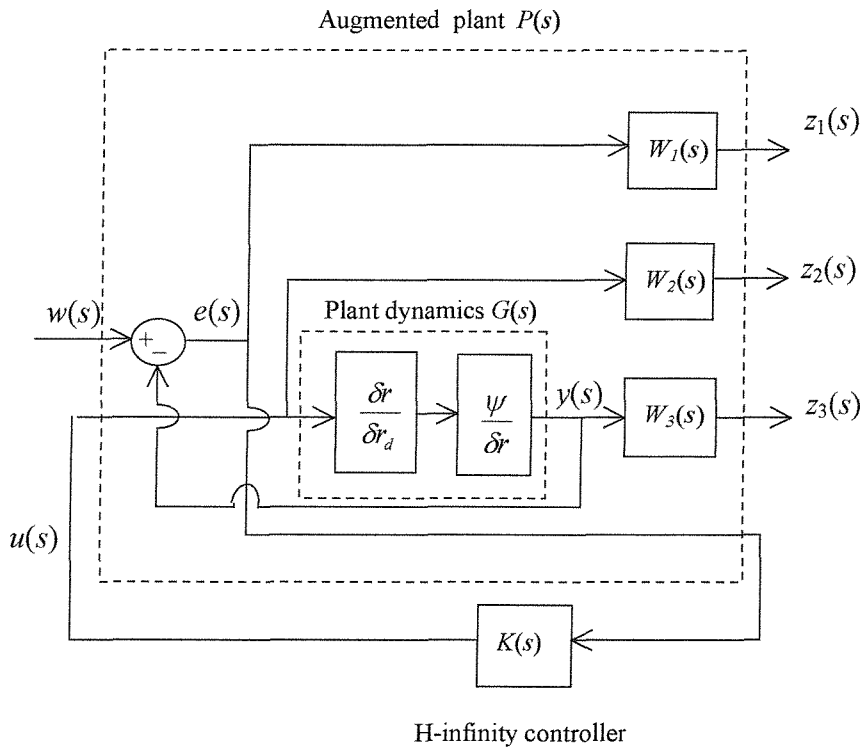


Figure 5.2: The arrangement to design a heading H-infinity control

Bilinear axis shifting transformation

Expression 5.7 above shows that there is a pole of $G(s)$ on the imaginary axis. This is undesirable because it can cause a marginally stable H-infinity controller. Therefore, a technique called the bilinear transform (axis shifting) was introduced to prevent this problem from occurring. A brief explanation of the bilinear axis shifting transformation is discussed in section 3.3.3. The theory is also discussed in [43].

Several tests using the bilinear axis shifting transform were carried out to find the suitable values for p_1 and p_2 where p_1 and p_2 are the endpoints of the diameter of a circle disk for the heading H-infinity controller. It was found that larger value of p_1 results in larger rudder deflection. These tests were done using the weighting function selected in (5.10) – (5.12). For example, with values of p_1= -0.3 and p_2 = infinity (∞), the rudder deflected to an angle of -18 degrees, while with values of p_1= -0.2 and p_2 = infinity (∞), the rudder deflected to an angle of -13 degrees. As a result, the time response was affected such that larger rudder deflection produces faster heading response. As for p_2, it was found that p_2 = infinity produced lower overshoot compared to other p_2 values such as 100 and $10e^5$. Although, p_2 = 100 and $p_2 = 10e^5$ resulted in faster rise time, a lower level of overshoot is preferable. This is because it is better to be slightly slower than to hit some obstacles along the way. In addition, the level of an overshoot in practice can be higher compared to the simulations carried in 5.5.3.1 – 5.5.3.5. As a result, the chosen values of p_1 and p_2 are –0.15 and ∞, respectively. With p_1 = -0.15, the rudder was deflected to an angle of –10 degrees. The command '*bilin*' from the MATLAB robust control toolbox was used to shift the poles, p_1 and p_2, from the original location ($j\omega$ - axis) in the s-plane to the $\tilde{s}$ - plane.

Weighting functions

The selections of some of the weighting functions are based on real data obtained by Lea from Haslar Tank tests with *Subzero II* [48]. The parameters obtained, such as the bandwidth of the relative heading error, were used to obtain the performance $W_1(s)$ and robustness $W_3(s)$ weighting functions.

Lea [1] had carried out some experimental trials for the heading control. The tests carried out were open loop tests. The experiments used the underwater vehicle, *Subzero II* described in section 2.1. Due to malfunction of the sensors, only two sets of data could be used for further investigation. Both data can be found in Appendix 8.4 and the plots in the time domain are shown in figures 5.3 and 5.4.

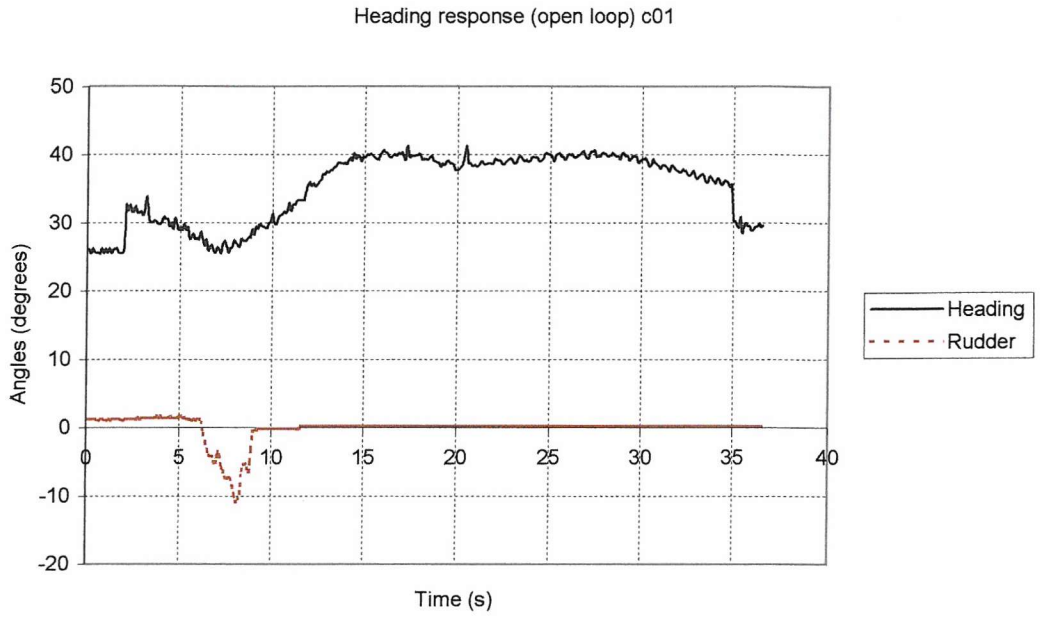


Figure 5.3: Time domain plot for data c01

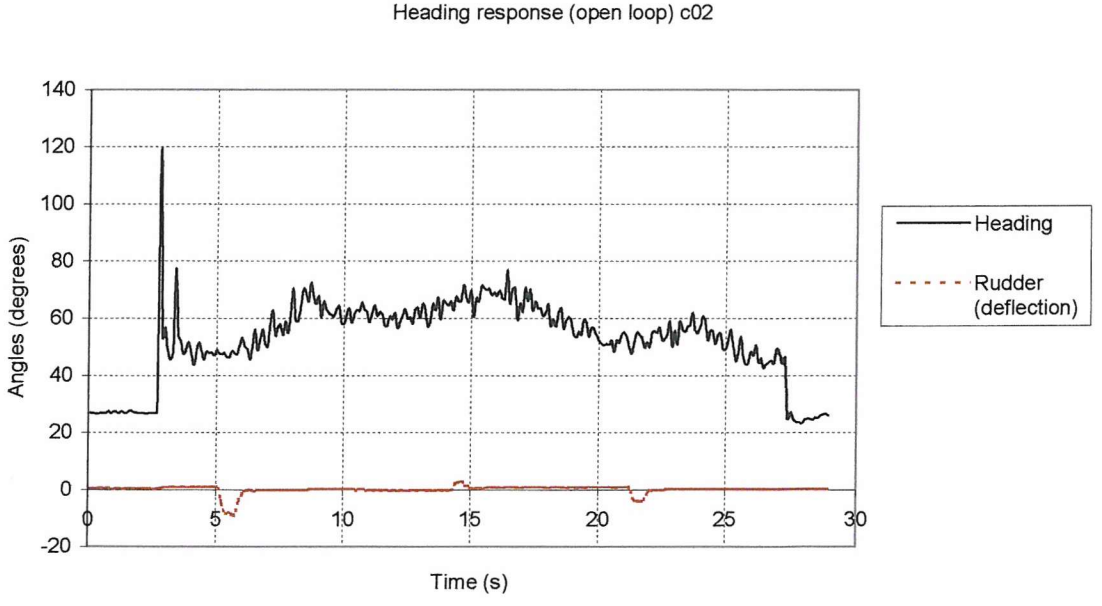


Figure 5.4: Time domain plot for data c02

In the open loop system experiments, the input to the heading subsystem was obtained from the rudder command or rudder deflection of the underwater vehicle. The demanded

rudder deflection is controlled by a joystick on shore. The aim of this project is to select the weighting functions based on real data. Since the performance weighting function $W_1(s)$ helps to attenuate the error of the heading subsystem, the heading error needs to be found. The heading error was considered as the uncertainty of the heading subsystem and disturbance was considered as one type of the uncertainties. The desired heading of the open loop experimental tests can be determined from the turn rate estimation. Generally, a turn rate tells us how much (in degrees) the underwater vehicle has turned in one second. Thus by knowing the value of the turn rate of the *Subzero II* relative to the rudder command (20 degrees) and the duration (in seconds) of the rudder deflects, the desired heading can be estimated based on the given turn rate value. Therefore heading angle was found using the equation:

Heading angle due to the demanded ruder deflection =
(Maximum turn rate estimation) x (time taken for the rudder deflection)

(5.8)

From Lea [1], the maximum turn rate value of *Subzero II* (without tether) was estimated to be 19.2 degrees per second. From figure 5.3, it is seen that the rudder took 1.1 seconds to turn to the desired heading (or to a settling point). In comparison, it took 0.6 seconds in the second set of data, *c02*. This is shown in figure 5.4.

Figures 5.3 and 5.4 show the time domain plot for data *c01* and data *c02*, respectively. The open loop system experiments ran for about 36 s for data *c01* and 29 s for data *c02*. The large initial jump starts for the first 7 s from figure 5.3 and 6 s from figure 5.4, are due to electromagnetic interference (EMI). The interference is caused by the running motor affecting the compass reading [1]. Therefore, these readings were ignored during the calculation for the desired heading angles below.

Data c01

Using formula (5.8) above, the estimated heading angle due to the rudder deflection is

19.7 degrees per second * 1.1 seconds = **21.67 degrees**

From figure 5.3 (ignoring the EMI effect), the underwater vehicle starts at a heading angle of 25.6 degrees. This heading angle was taken as the starting point (0 degrees). Therefore, data c01 in Appendix 8.4.1 need to be subtracted by 25.6 degrees.

Data c02

From formulae (5.8), the estimated heading angle due to the rudder deflection is

19.7 degrees per second * 0.6 seconds = **11.87 degrees**

From figure 5.4, it is seen that the underwater vehicle starts from a heading angle of 47.9 degrees, after 6s. As above (*data c01*), the 47.9 degrees point was taken as the starting point. Thus, the measured data *c02* was subtracted by 47.9 degrees.

The error of the heading can be estimated from the resulting desired heading angles obtained above by taking the difference between the measurement (actual) heading angles and desired heading angles.

Frequency response of the heading error

All the results above have been carried out in the continuous time domain. To compute an H-infinity controller, the weighting functions need to be specified in *s*-plane form. To do this, the continuous time domain data is transformed to the frequency response domain using the Fast Fourier Transformation (FFT). The transformation was carried out using the MATLAB robust control toolbox using the '*fft*' command. For further investigation, only

the steady state regions were taken to obtain the frequency response of the heading error. This refers to when the underwater vehicle has settled down, after few seconds in a run. The steady state region for figure 5.3 and figure 5.4 was taken between 15 s and 30 s and between 8 s and 18 s, respectively. From observations, the plot of the weighting function is always in dB versus rad/s form. Therefore, a decibel (dB) unit is required for the vertical axis of the frequency response. The relative error was used to calculate the heading error in decibels (dB) as shown in (5.9) below: -

$$(\text{dB}) = 20 * \log_{10}(\text{ratio})$$

$$(\text{dB}) = 20 * \log_{10}(\text{relative heading error}) \qquad \textbf{(5.9)}$$

The relative heading error is defined as: -

$$\textit{Relative heading error} = \frac{\textit{Actual heading - Desired heading angle}}{\textit{Desired heading angle}}$$

$$\textit{Relative heading error} = \frac{\textit{Heading error}}{\textit{Desired heading angle}}$$

The full program listing of the transformation from time domain to frequency domain for the heading error can be found in Appendix 8.4. The results of the transformation into the frequency domain are shown in figures 5.5 and 5.6 below. Figures 5.5 and 5.6 show the frequency response of the heading error for data *c01* and data *c02*, respectively. The vertical axis for the frequency response plots is in decibel (dB) while the horizontal axis is in radian/ seconds (rad/s).

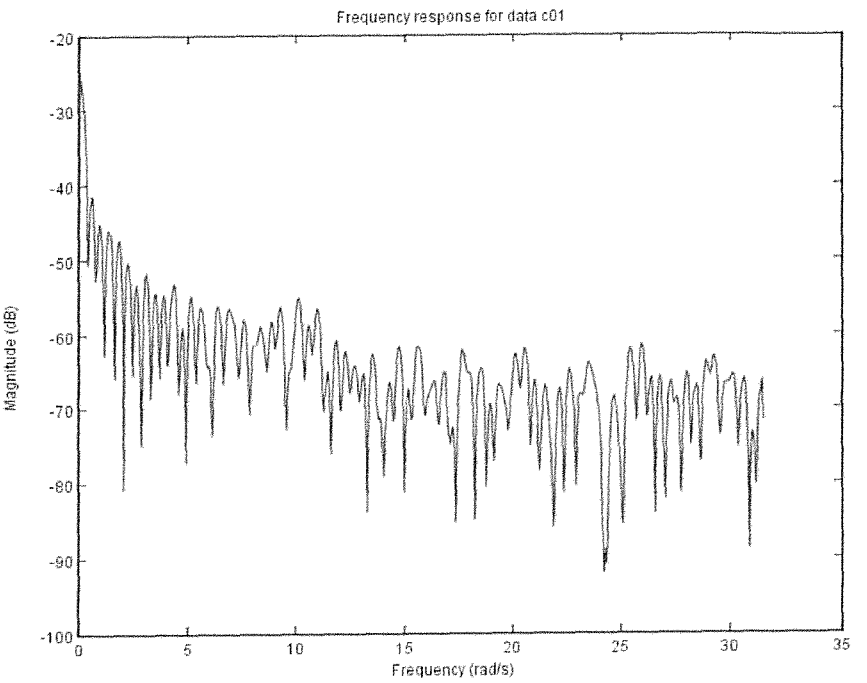


Figure 5.5 Frequency response of the relative heading error for data *c01*.

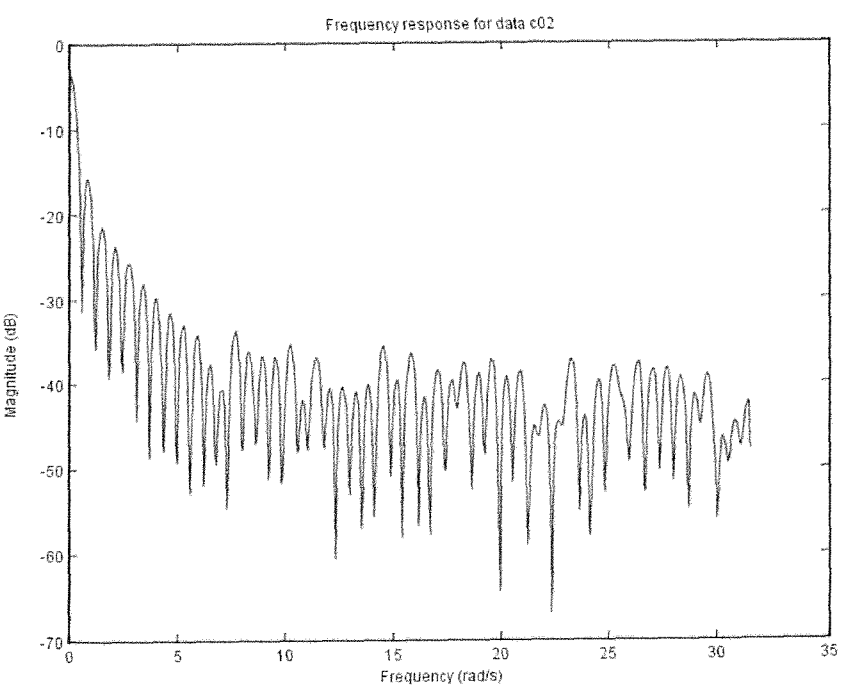


Figure 5.6: Frequency response of the relative heading error for data *c02*.

From figures 5.5 and 5.6 above, the frequency of the maximum magnitude for the relative heading error is 0.06 rad/s. From findings [47], it was found that the frequency of the marine environment is very low and the values are usually below 0.3 rad/s. Therefore, the value of frequency obtained from data *c01* and *c02*, 0.06 rad/s, is a reasonable frequency for the heading error.

From the experimental results obtained by Lea [1], the frequency for the high frequency disturbances, which is the sensor noise, is about 10 rad/s. Without the sensor, the noise produced was about 13 rad/s. Therefore, the acceptable bandwidth for noise is at least 10 rad/s and therefore, the gain for the frequency response needs to be small in the high frequency range to ensure sufficient noise rejection.

From the maximum frequency of the heading error, bandwidth of noise as well as the closed loop objectives, the chosen weighting functions are listed below: -

$$W_1(s) = \frac{5.623s + 1}{10s + 1e^{-5}} \tag{5.10}$$

$$W_2(s) = 1 \tag{5.11}$$

$$W_3(s) = \frac{s^2 + 0.06325s + 0.001}{0.001667s^2 + 0.8165s + 100} \tag{5.12}$$

The performance weighting function, $W_1(s)$ is to assist the sensitivity function $S(s)$ with disturbance rejection. The control weighting function $W_2(s)$ is included to obtain an optimal control system. The robustness weighting function $W_3(s)$ is added to minimise the complementary sensitivity function $T(s)$ at the high frequency, in order to reduce the noise

effect in the system. The individual frequency response plots are shown in figure 5.7 and explanation on the selection of each weighting function is discussed below.

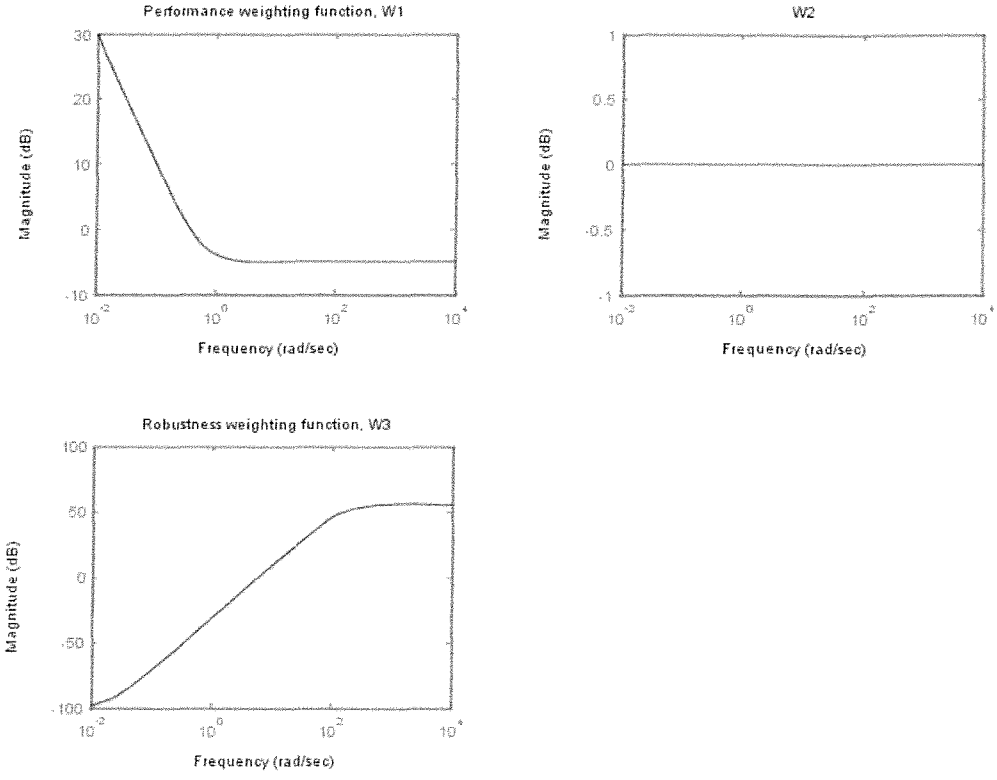


Figure 5.7: Frequency response plots of each weighting function

Performance weighting function $W_1(s)$

The concentration of underwater disturbances such as current velocity lie between 0 – 1.5 rad/s [42]. The maximum relative heading error obtained in figures 5.5 and 5.6 was 0.06 rad/s but for simplicity, a value of 0.1 rad/s was assigned as the bandwidth for the performance weighting function. Larger bandwidth may affect the stability margin of the sensitivity of the heading subsystem. A high gain of 30 dB and above is given to the low frequency range between 10^{-2} and 10^{0} rad/s, to ensure that the heading subsystem of the

underwater vehicle is not affected by any disturbances underwater. The gain at the high frequency range is insignificant because the performance weighting function $W_1(s)$ concentrates in rejecting disturbance in the low frequency range. Since, the magnitude value at the high frequency is not directly related to $W_1(s)$, a gain of –5dB was chosen.

Control weighting function $W_2(s)$

From expression (3.5), both disturbances $d(s)$ and noise $n(s)$ are to be kept low below 0 dB to ensure that both the $d(s)$ and $n(s)$ signals are rejected. This is done by setting $W_2(s) = 1$, to ensure that the control weighting function $R(s)$ is below 0 dB, in both low and high frequencies regions. From several tests carried out with $W_1(s)$ and $W_3(s)$ as in (5.10) and (5.12), higher constant values of $W_2(s)$, for example $W_2(s) = 1.5$, produced lower singular value of T_{zw} (close to 0 dB). In comparison, lower constant values of $W_2(s)$, for example $W_2(s) = 0.5$ produced higher singular value of T_{zw} (further away from 0 dB). These values of $W_2(s)$ resulted in overshoots ranging between 9 and 12 degrees. Therefore, $W_2(s) = 1$ was chosen because when rudder delay of 0.33s was included into the simulation, the overshoot value was low (1.1 degrees). The plot for singular value of T_{zw} with $W_2(s) = 1$ is found in figure 5.12 and the heading response is shown in figure 5.18.

Robustness weighting function $W_3(s)$

$W_3(s)$ is used to reduce the noise contamination in the high frequency region. This can be achieved by having a large value for the complementary sensitivity function $T(s)$ in the high frequency range. The bandwidth of the noise was found earlier to be 13 rad/s but a bandwidth of 10 rad/s was assigned in designing the robustness weighting function. A lower frequency value of 10 rad/s was chosen so as to take into account more lower frequency uncertainties as the bandwidth for uncertainty (disturbance) is very low; of the order of 0.1 rad/s. Examples of underwater noise are ambient noise (movement of icebergs): 25 rad/s, shipping traffic noise (125.7 - 1885.2 rad/s), seismic blast (6.3×10^3 – 9.4×10^3 rad/s) [49]. A gain of 55.6 dB and above was estimated in the high frequency

region, as to attenuate the level of noise into the system. A low frequency gain was estimated as -100 dB.

5.3.2 H-infinity controller

There are two methods available in the MATLAB robust control toolbox, to compute an H-infinity controller. This is done by using the commands *hinf* and *hinfopt*. *hinf* finds a stabilising controller $K(s)$ for a system by solving the small gain infinity-norm robust control problem, such that $\| T_{zw} \|_\infty < 1$. T_{zw} is the closed loop transfer function between the output, z and input, w of the augmented system as shown in figure 3.7 and is represented as the linear fractional transformation as $F_l(P, K) = P_{11} + P_{12}K (I-P_{22}K)^{-1} P_{21}$. The optimal H-infinity control law is discussed in section 3.3. *hinf* implements the 'loop-shifting' two Riccati formulae for the infinity-norm control. In comparison, the *hinfopt* command, computes the optimal H-infinity controller using the loop-shifting formulae of *hinf* via γ-iteration such that $\| \gamma T_{zw} \|_\infty < 1$. Therefore, for an optimal H-infinity controller, γ needs to be below 1.

Several simulation tests were carried out using both '*hinf*' and '*hinfopt*' commands on heading control of the underwater vehicle *Subzero II*. It was found that the resulting gamma γ from *hinfopt* was 1.5234 where as the gamma γ value used in the *hinf* command was 1. The tests were carried out using the weighting functions in (5.10 - 5.12). In addition, the use of *hinfopt* command produced unstable rudder deflection for the first 2s of the simulation run. When the random noise was added into the simulation, the resulting rudder deflection was very noisy. In comparison, the use of the *hinf* command resulted in better robust analysis results as shown in section 5.4 and 5.5. Therefore, the *hinf* command was preferred because from inequality $\| T_{zw} \|_\infty < \gamma$, gamma γ needs to be a minimal value. Thus, the command *hinf* was used to solve the H-infinity control for the heading control of an underwater vehicle.

From the Bilinear axis shifting transformation in section 3.3.3, the final pole of the H-infinity controller needs to be shifted back to the s-plane by using the inverse Bilinear

transformation (shifting transformation). For simplicity the formula of the inverse Bilinear transformation is as below: -

$$\tilde{s} = \frac{-s + p_1}{\frac{s}{p_2} - 1}$$

The resulting H-infinity controller $K(s)$ obtained from MATLAB robust control toolbox in the s-plane is shown below. It is a 7th order state controller in the s-plane because the total number of order from the augmentation process (plant + weighting functions) is 4 + (1 + 0 + 2) = 7.

$$K(s) = \frac{\partial r_d}{\psi} =$$

$$\frac{-0.6595\, s^6 - 335.6\, s^5 - 4.572e^4\, s^4 - 7.725e^5\, s^3 - 4.072e^6\, s^2 - 5.389e^6\, s - 9.017e^5}{s^7 + 510.9\, s^6 + 7.037e^4\, s^5 + 1.314e^6\, s^4 + 8.643e^6\, s^3 + 2.217e^7\, s^2 + 1.935e^7\, s + 2.433e^6} \quad \textbf{(5.13)}$$

5.3.3 Discretisation

The underwater vehicle simulation program, *AutoROV*, was used to test the resulting H-infinity controller for the heading control. Because the program only handles digitised controllers, the expression 5.13 has to be digitised. There are a few discretisation methods available in the MATLAB robust control toolbox such as zero-order hold, first-order hold and bilinear approximation. In this work, the Tustin approximation (Bilinear transformation) is used because it preserves the H-infinity norm. In addition, it only involves a simple substitution of function z for s to get $K(z)$ by using the formula (5.14)

below. Furthermore, it maps the entire s-plane to the z-plane, thus preventing any frequency domain aliasing problems [46].

Therefore, the digitised H-infinity controller of (5.12) above is determined by simple substitution of formula (5.14) into (5.13) giving the final digitised H-infinity controller $K(z)$ shown in (5.16).

$$F(z) = F(s) \mid \frac{2}{T}\left(\frac{z-1}{z+1}\right) \tag{5.14}$$

Where T is the sampling time.

There are two reasons for choosing the sampling time to be 0.1s.

1) To prevent the problem of aliasing

For better sampling, the sampling frequency, which is the reciprocal of the sampling time $f_s = \frac{1}{T_s}$, has to be at least twice the highest frequency in the signal. It was found in section 5.3 that the estimated highest frequency for noise is 13 rad/s or 4.14 Hz. Therefore, to avoid aliasing, the sampling frequency was chosen to be 10 Hz. Thus, the sampling time of 0.1s.

2) To preserve the result of the continuous H-infinity control

One condition for the Tustin approximation to preserve the result of the continuous H-infinity is to have the sampling time several times higher than the bandwidth of the heading control. From Lea [1] and Logan [9], the bandwidth of the heading control was estimated to be 1 Hz. To preserve the continuous H-infinity control, a sampling frequency of 10 Hz was chosen and this is equivalent to a sampling time of 0.1s.

After the transformation from the continuous to the discrete signal using the Tustin approximation, the resulting digitised H-infinity controller is

$$K(z) = \frac{\partial r_d}{\psi} =$$

$$\frac{-0.02985\, z^7 - 0.001576\, z^6 + 0.06648\, z^5 - 0.001811\, z^4 - 0.04793\, z^3 + 0.00625\, z^2 + 0.01119\, z - 0.002969}{z^7 - 1.753\, z^6 - 0.5098\, z^5 + 2.37\, z^4 - 0.8266\, z^3 - 0.6855\, z^2 + 0.4863\, z - 0.08074}$$

(5.15)

In time delay (z^{-1}) form (divide each term by z^7) [30], the expression (5.15) becomes

$$K(z) = \frac{\partial r_d}{\psi} =$$

$$\frac{-0.02985 - 0.001576\, z^{-1} + 0.06648\, z^{-2} - 0.01811\, z^{-3} - 0.04793\, z^{-4} + 0.00625\, z^{-5} + 0.01119\, z^{-6} - 0.002969\, z^{-7}}{1 - 1.753\, z^{-1} - 0.5098\, z^{-2} + 2.37\, z^{-3} - 0.8266\, z^{-4} - 0.6855\, z^{-5} + 0.4863\, z^{-6} - 0.08074\, z^{-7}}$$

(5.16)

δr_d is the demanded deflection of rudder and ψ is the vehicle global yaw angle.

5.4 Robust analysis

In this section, the results of the inequalities (5.1) - (5.3) and other closed loop objectives are presented. These results were obtained using the MATLAB robust control toolbox.

Norm which is denoted by the symbol || ||, can be defined as an overall measure of the size of a matrix, a signal, a matrix or a system [31]. There are many definitions for and it is dependent to whether the element is a matrix, a vector, a signal or a system. In this project, the infinity norm of a system is required. An infinity norm of a system or a transfer function, $F(s)$ is the H-infinity norm and it is the peak value of the maximum singular value $F(s)$. This infinity norm can be denoted as $\|F(s)\|_\infty = \max_\omega \bar{\sigma}(F(j\omega))$. Refer section 3.2 and [31] for further explanation for infinity norm. As mentioned in section 3.1, singular value or spectral norm is defined as the positive square roots of the eigenvectors of $Q^T Q$, where Q is a complex matrix. The singular value is denoted with the symbol σ. Further discussion on the terms: norm, infinity norm and singular value, can be found in [31].

5.4.1 Disturbance rejection

For disturbance rejection, the sensitivity weighting function must be less than the inverse of the performance weighting function such that $S(s) < W_1(s)^{-1}$ [31], [37]. From figure 5.8 below, it is seen that this is the case. Thus the disturbance is guaranteed to be rejected in the heading subsystem.

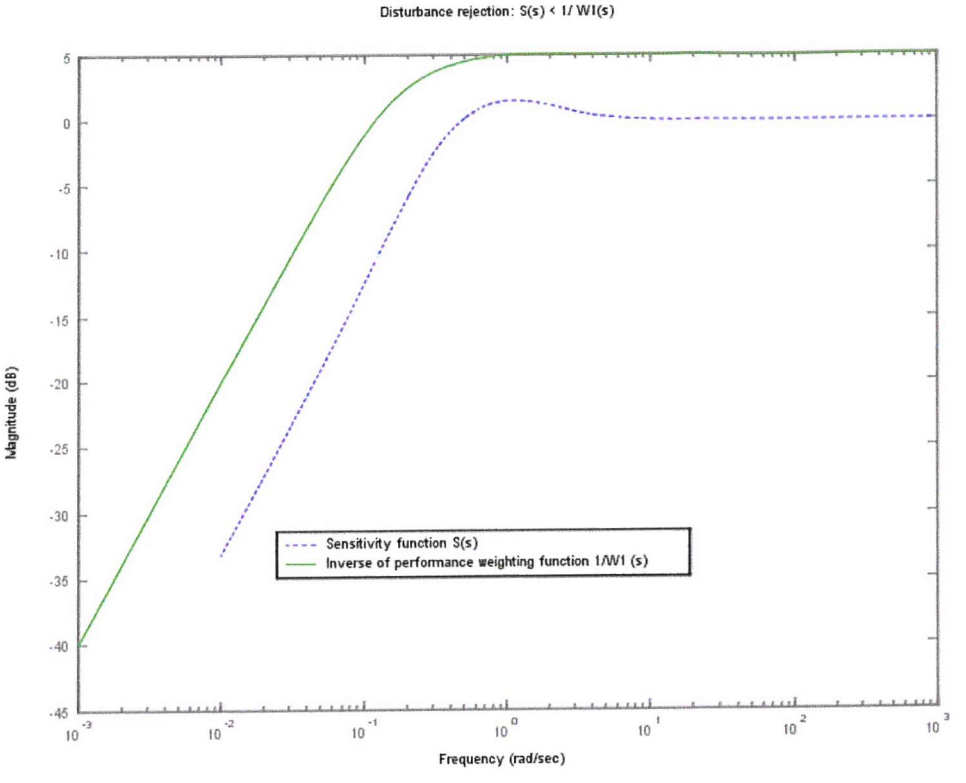


Figure 5.8: Disturbance rejection: $S(s) < W_1(s)^{-1}$

In this case, the sensitivity function relates the actual heading and the disturbances or relative heading error such that

$$S(s) = \frac{\psi}{\psi_e} = \frac{\psi}{d}$$

The frequency of the relative heading error or disturbances of the underwater environment was estimated to be 0.1 rad/s as found in section 5.3.1. From the figure above, the sensitivity of the heading subsystem at that frequency is – 12.5 dB.

$$-12.5 \text{ dB} = 10^{(-12.5/20)} = 0.237$$

From (3.3),

$$y(s) = S(s) \times d(s)$$
$$\psi(s) = 0.237 d(s)$$

This indicates that the disturbance of the marine environment is attenuated by about 0.24 for the heading response of the underwater vehicle.

5.4.2 Noise attenuation

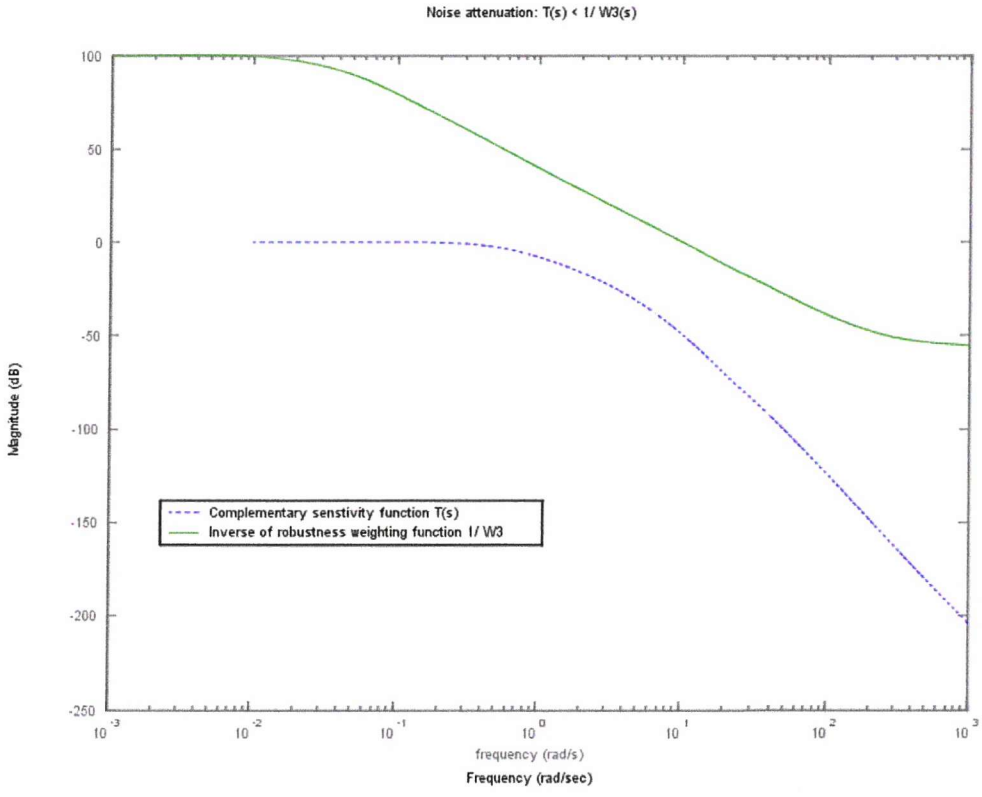


Figure 5.9: Noise attenuation: $T(s) < W_3(s)^{-1}$

For a system to be robust from noise, it is required that the complementary sensitivity function is less than the inverse of the robustness weighting function such that $T(s) < W_3(s)^{-1}$. From figure 5.9 above, the curve $T(s)$ is below the inverse of $W_3(s)$. This indicates that the condition for noise attenuation is achieved.

In section 5.2.3, one of the robustness specifications was for the $T(s)$ curve to have a roll-off rate of at least - 20 dB/ decade. This is to guarantee noise attenuation in the system. Figure 5.9 shows that the roll-off rate is - 75 dB/ decade, which should ensure good noise rejection. Further noise reduction at the high frequencies can be fulfilled by having low negative magnitude at the frequency of the noise of the system. From the robustness specification, it is required that the magnitude of the complementary weighting function $T(s)$ to be at least –20 dB at 10 rad/s. From figure 5.9, the magnitude is about – 49 dB at 10 rad/s. Hence this robustness specification is fulfilled.

5.4.3 Reference tracking

One of the closed loop objectives in section 3.1 is for the system to have the ability to track the changing desired output and in this case the desired heading angles. In order for this to happen, it is required that the complementary sensitivity function $T(s)$ is below 1 or 0 dB in magnitude at low frequency. From figure 5.9 above, the complementary sensitivity function $T(s)$ is ~ 0 dB in magnitude at the low frequency range until 0.2 rad/s. From this result it indicates that the heading subsystem should be able to track the changing desired heading angle until frequency 0.2 rad/s.

5.4.4 Optimal control

From section 5.2.2, it is required that the signals of both disturbances and noise are kept low. As mentioned earlier, the disturbances signal concentrates at the low frequency region where as the noise signal is mainly distributed at the high frequency region.

Therefore, it is essential to keep the low frequency as well as the high frequency regions small in magnitude (dB). In addition from (5.2), an optimal control of the rudder can be obtained if the control sensitivity function is less than the inverse of the control weighting function such that $R(s) < W_2(s)^{-1}$.

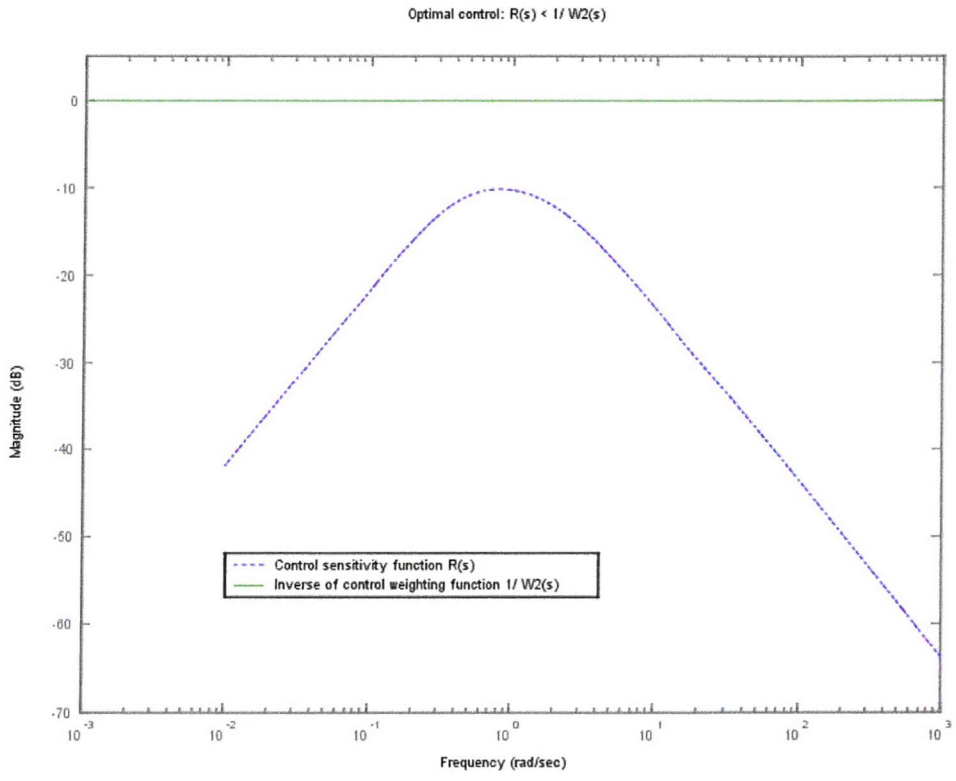


Figure 5.10: Optimal control: $R(s) < W_2(s)^{-1}$

As described earlier, the control weighting function, $W_2(s)$ has been set to 1 as shown in expression 5.11 in section 5.3.1. In figure 5.10 above, the resulting control sensitivity function is below the 0 dB line for both low and high frequencies regions. Therefore, the condition for optimal control is fulfilled.

The specification for optimal control on the actuator, in this case the rudder, can also be examined from the frequency response of the controller. In order to obtain the optimal control of the rudder, the magnitude of the controller needs to be low at high frequencies. The frequency response of the controller is shown in figure 5.11 below. It is observed that the controller generally has a low magnitude (below 1 or 0dB) for all frequencies. Thus, the condition for the optimal control for the rudder is ensured.

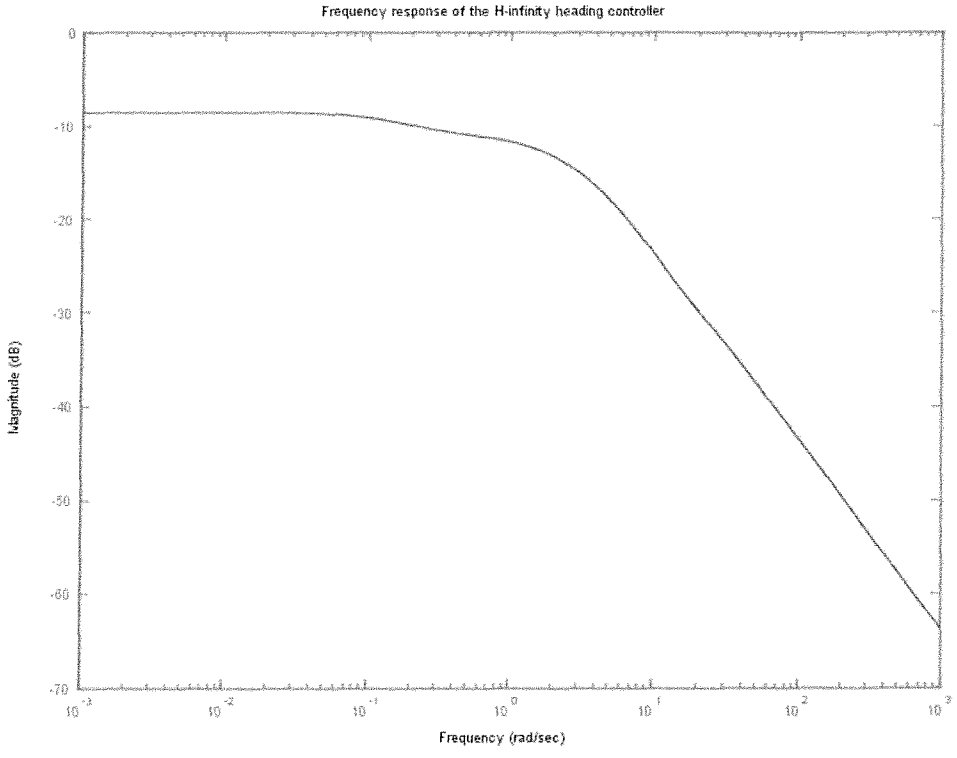


Figure 5.11: Frequency response of the H-infinity controller

5.4.5 Closed loop transfer function, T_{zw} (Controller stability)

T_{zw} is the transfer function relating the input and the output of the closed loop system. For a controller to be stable, it is required that the infinity norm of T_{zw}, is less than a positive constant, γ, $\| T_{zw} \|_\infty < \gamma$ and γ was chosen as 1. The reason of choosing γ=1 is discussed in section 5.3.2.

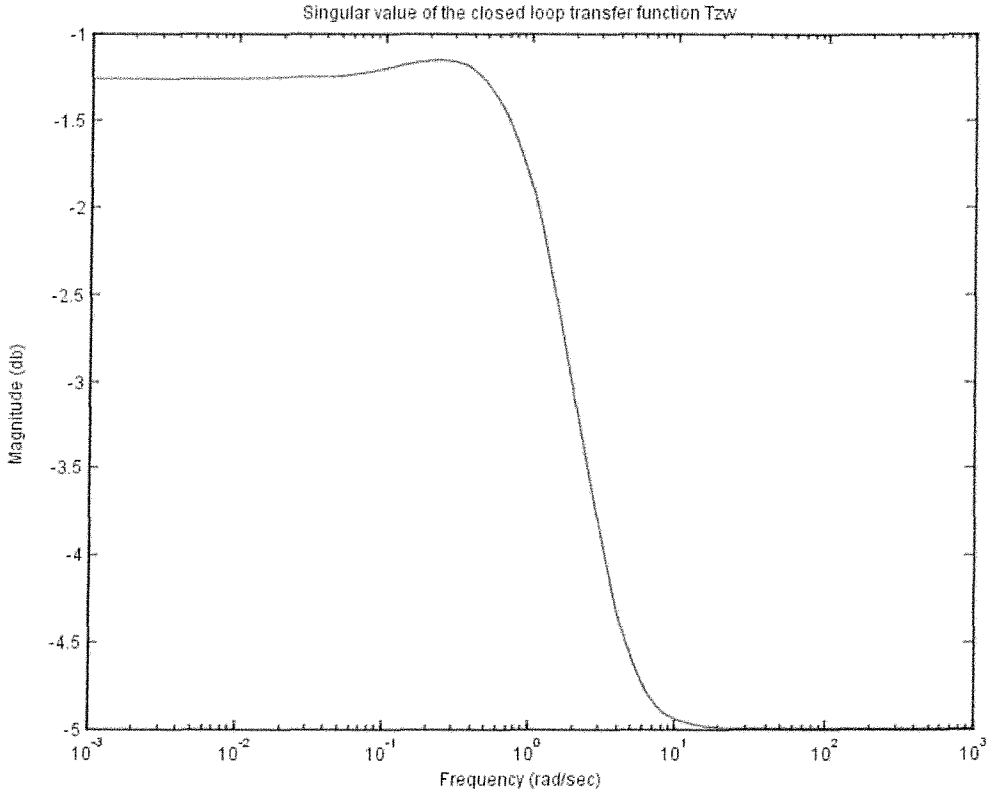


Figure 5.12: Singular value of the closed loop transfer function T_{zw}

Figure 5.12 shows the maximum singular value of the closed loop transfer function T_{zw} and $\|T_{zw}\|_\infty = \max_\omega \bar{\sigma}(T_{zw}(j\omega))$. For a stable H-infinity controller to exist, $\| T_{zw} \|_\infty < 1$. From

the plot above, T_{zw} is below 1 or 0 dB for all frequencies. Therefore, the designed H-infinity for the heading subsystem is stable.

The robust analysis above is from modern control, where the singular value is used as a measure of robustness. Below is a robust analysis from the classical approach which is based upon the stability margins: gain and phase margins.

5.4.6 Stability margin

Stability margin is a form of stability robustness of a system [37]. It is a measure of robustness on the stability of the system with respect to the variation in the system's model. It can either be measured using the Bode plot of $G(j\omega)\ K(j\omega)$ or the Nyquist plot of $G(s)K(s)$. Stability margins consists of gain margin and phase margin.

The expression for a closed loop system is $\frac{C(s)}{R(s)} = \frac{G(s)}{1+G(s)K(s)}$. The closed loop becomes unstable when $\frac{C(s)}{R(s)} \rightarrow \infty$ which can be either when:

1) $G(s) \rightarrow \infty$ or

2) $1 + G(s)K(s) = 0 \Rightarrow G(s)K(s) = -1$

In the complex plane,

$$G(j\omega)K(j\omega) = (-1, j0)$$
$$= 1 \angle -180^0.$$

The above symbol indicates that a system is unstable if the magnitude of the open loop gain exceeds unity at a phase lag of 180 degrees. For this project, the Bode plot is used to obtain the gain and phase margins. The gain margin tells us how far away in dB the

system is, before it goes unstable. From the plot, the gain margin is the gain of $G(j\omega)$ $K(j\omega)$ when its phase crosses the –180 degrees phase line. Similarly, the phase margin tells us how many degrees before the system becomes unstable. The phase margin is the difference between the phase of $G(j\omega)$ $K(j\omega)$ and –180 degrees, when the gain of $G(j\omega)$ $K(j\omega)$ crosses the 0dB line. For simplicity, both margins are shown in the figure below. Negative values of gain and phase margins indicate an unstable system.

Plant only

From figure 5.13 below, the gain margin is found to be infinity dB while the phase margin is (75.9 + 180) = 255.9 degrees at 1.27 rad/s.

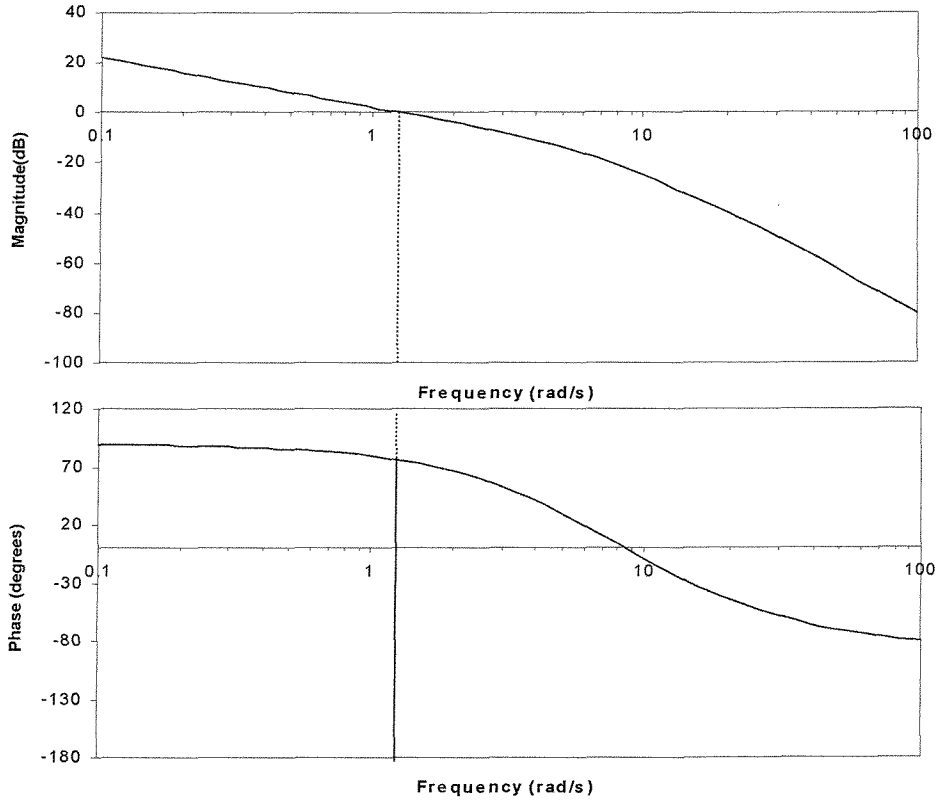


Figure 5.13: Stability margin for the plant only

Open loop transfer function, G(s)K(s)

Figure 5.14 below shows the stability margin of the open loop *G(s)K(s)* heading control system. The resulting gain margin is 23.3 dB at 3.05 rad/s, where as the phase margin is found to be 71.4 degrees at 0.37 rad/s. Although both gain and phase margins were reduced, the margins are still large before the system goes unstable. The minimal gain and phase margins required before a system goes unstable is 2 (6dB) and between 30 degrees and 60 degrees, respectively [31], [43].

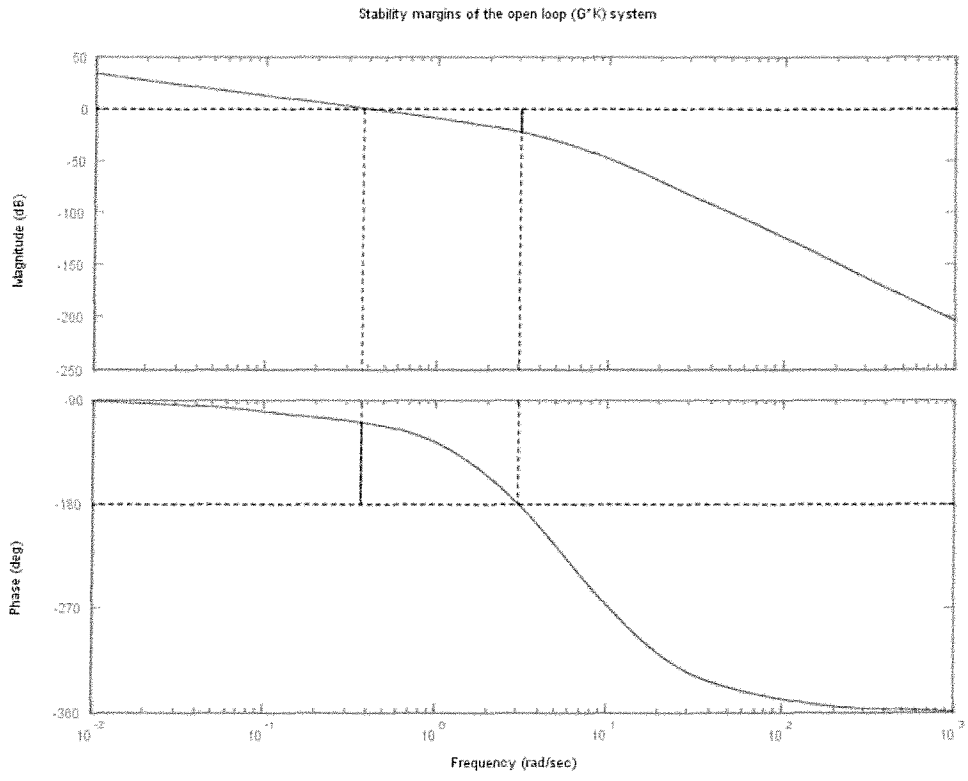


Figure 5.14: Stability margin of the open loop system (*G(s)K(s)*)

In general, the H-infinity controller fulfilled most of the closed loop objectives stated in sections 3.1 and 5.2. The H-infinity controller was then applied to a model of an

underwater vehicle to obtain several heading responses using the underwater vehicle simulation package, *AutoROV*, and this is described next.

5.5 Simulation results

Several simulation tests were carried out using *AutoROV* and the heading response of the H-infinity control were compared to the PID control designed by Lea [1]. A brief description of *AutoROV* was given in section 4.2. For all the tests, the time delays for the motor and sternplane were set to 0.15s and 0.23s, respectively. These delays are mainly due to sensors and the communication link. The desired heading angle was set at 40 degrees. The speed of the motor is 1.3 m/s.

In *AutoROV*, the six degree of freedom equations of motion are catagorised into three subsystems: speed, heading and depth controls. From Lea [1],

PI speed control:

$$m_d = 2000u_e + 3200\sum_0^k \frac{u_{e_k}}{T} \tag{5.17}$$

where m_d is the motor command, u_e is the speed error and $\sum_0^k \frac{u_{e_k}}{T}$ is the speed integrator.

Several heading simulation tests were carried out to find the best value for the proportional and integrator gains. As a result, the values of 2000 for the proportional gain and 3200 for the integrator gain, gave good heading responses. The transfer function form (not including time delay) is as below.

$$\frac{\Delta u_e}{\Delta m_d} = \frac{2.9\times10^{-3}}{(s+4.17)(s+0.5)}$$

PID heading control:

$$\delta r_d = -0.6\psi'_k - 0.05T\sum_{n=1}^{n=k}\psi'_n - 0.1\dot{\psi}' \qquad \textbf{(5.18)}$$

where δr_d is the demanded rudder deflection and $\psi' = \psi_d - \psi$ is the heading error. The transfer function form for the PID heading control can be found in expression (5.4).

PID depth control:

Two controllers were used for the depth control: depth-pitch controller at the outer loop and pitch-sternplane controller at the inner loop. The outer loop produces the commanded pitch value while the inner loop gives the demanded sternplane value based on the pitch error. This arrangement is used because the value of the commanded pitch (from the outer loop) can be limited below 90 degrees to avoid singularities problem. In addition, the dynamics of pitch-sternplane is very similar to the dynamics of heading-rudder in the heading subsystem [1]. A suitable PID controller for the heading subsystem had already been designed and therefore, the time to design the pitch-sternplane controller can be reduced.

depth-pitch controller

$$\theta_d = -0.5z' - 0.05T\sum_{n=1}^{n=k} z'_n - 0.1\dot{z}' \qquad \textbf{(5.19a)}$$

where θ_d is demanded pitch angle and $z' = z_d - z$ is the depth error.

pitch-sternplane controller

$$\delta s_d = -0.8\theta' - 0.05T\sum_{n=1}^{n=k}\theta'_n - 0.3q \qquad \textbf{(5.19b)}$$

where δs_d is demanded sternplane and $\theta' = \theta_d - \theta$ is the pitch angle error. q is the pitch rate. In transfer function form:

$$\frac{z}{\delta s} = \frac{-1.07(s-9.6)(s+2.2)}{s(s+0.081)(s+1.8)(s+8.4)}$$

where T is the sampling time and is assumed to be 0.1s.

These are the three controllers (5.17 - 5.19) used in the *AutoROV* underwater simulation program to obtain several PID heading responses. The controllers are discussed in greater detail in Lea [1]. As for the H-infinity heading control, the H-infinity heading controller is given in (5.16) for the heading subsystem. For the speed and depth controls, the PI speed in (5.17) and PID depth in (5.19) controllers, respectively were used. The program listing used to obtain the heading responses shown in figures 5.15 - 5.19, can be found in Appendix 8.6.

There are three different types of uncertainties added into *AutoROV* [12]. They are current velocity as disturbances, sensor noise and time delay. The effect of these uncertainties are investigated below. A brief description of the uncertainties is given in the following section.

5.5.1 Uncertainties in *AutoROV*

a) Disturbances

The disturbances included in the simulation package, *AutoROV* are water current velocity. These disturbances were estimated by Lea [12]. The water current velocity disturbances in (5.20) and (5.21) below are added to the axial (x-axis) and lateral (y-axis) forces, respectively. Because the underwater environment is highly coupled, the heading response is also affected. The expressions for the disturbances at their respective coordinates are:

At *axial coordinates*: $-0.1 \cos(0.2t) \sin(\psi)$;

(5.20)

At *lateral coordinates*: $-0.1 \cos(0.2t) \sin(\pi/2 - \psi)$;

(5.21)

b) Random noise

The heading angle is measured using the digital module compass. According to Lea [1], the motor power cables and the battery packs produced an electromagnetic interference on the compass readings. In the underwater vehicle simulation program, *AutoROV*, random noise having a range between - 2 degrees and 2 degrees is added to the compass data based upon observations from the underwater vehicle data in Lea [1].

c) Time delay

There are some rudder time delays in the heading subsystem. From Lea [1], the longest delay is due to the digital module compass with a delay of 0.2s. The rudder is structured such that there are two rudder surfaces and they are linked together to form a single rudder. In addition, there is also delay in the rudder servo. The worst case is when the rudder moves across the centre of the tail section resulting a rudder delay of 0.13s. Therefore, the total worst case delay due to sensors and the rudder is 0.33s.

5.5.2 Simulation runs

Six simulation runs using the H-infinity heading controller were carried out. The digitised H-infinity heading controller (5.16) was then compared to the PID controller designed by Lea [1].

a) No uncertainty is added.

All the parameters in the option file, such as the disturbances and noise, were ignored (FALSE). The time delay for the rudder was set to 0.01s. The rudder delay cannot be set to zero because it resulted in a numerical error. The results of the heading response with these settings are shown in Figure 5.15 below.

In following simulation runs (b) and (c), the RUD_TIME_DELAY were set to 33 and 70, respectively. The parameters: DISTURBANCES and SENSOR_REAL were set to TRUE. The others parameters in the *option.h* file were kept as FALSE. The motor and sternplane time delays are set as 0.15s and 0.23s, respectively to create a realistic situation.

b) 3 uncertainties were added: disturbances, random noise and rudder delay of 0.33s.

The simulated heading response is shown in figure 5.16.

c) 3 uncertainties were included: disturbances, random noise and rudder delay of 0.70s.

This simulation run was done to test if *Subzero II* could withstand the pure delay of the heading subsystem. The resulting H-infinity heading response is found in figure 5.17.

The following simulation runs (d) and (e), the RUD_TIME_DELAY were set to 33 and 70, respectively. The parameter SENSOR_REAL was set to TRUE. The others parameters in the *option.h* file were kept as FALSE. The MTR_TIME_DELAY and SPL_TIME_DELAY are set as 15 and 23, respectively.

d) 2 uncertainties were added: random noise and rudder delay of 0.33s.

The result of the simulated H-infinity heading response is shown in figure 5.18.

e) 2 uncertainties were included: random noise and rudder delay of 0.70s.

The simulated H-infinity heading response can be obtained in figure 5.19.

For simplicity, the simulation runs are tabulated in Table 5.1. Other parameters are kept as FALSE. The time delay for the motor and sternplane are 15 (0.15s) and 23 (0.23s), respectively.

Parameters / Simulation runs	Disturbance	Random noise	Rudder time delay
a)	FALSE	FALSE	1 (0.01s)
b)	TRUE	TRUE	33 (0.33s)
c)	TRUE	TRUE	70 (0.7s)
d)	FALSE	TRUE	33 (0.33s)
e)	FALSE	TRUE	70 (0.7s)

Table 5.1: Simulation runs

f) Nonlinear underwater vehicle simulator

The H-infinity control was also tested in Feng's underwater simulation program, *Subzero* [24]. It was then compared to the simulated PID control which was also simulated with the *Subzero* simulation program. The result of the simulation from Feng [24] is shown in figure 5.20.

5.5.3 Simulation results

Using the AutoROV simulation program

5.5.3.1 Ideal

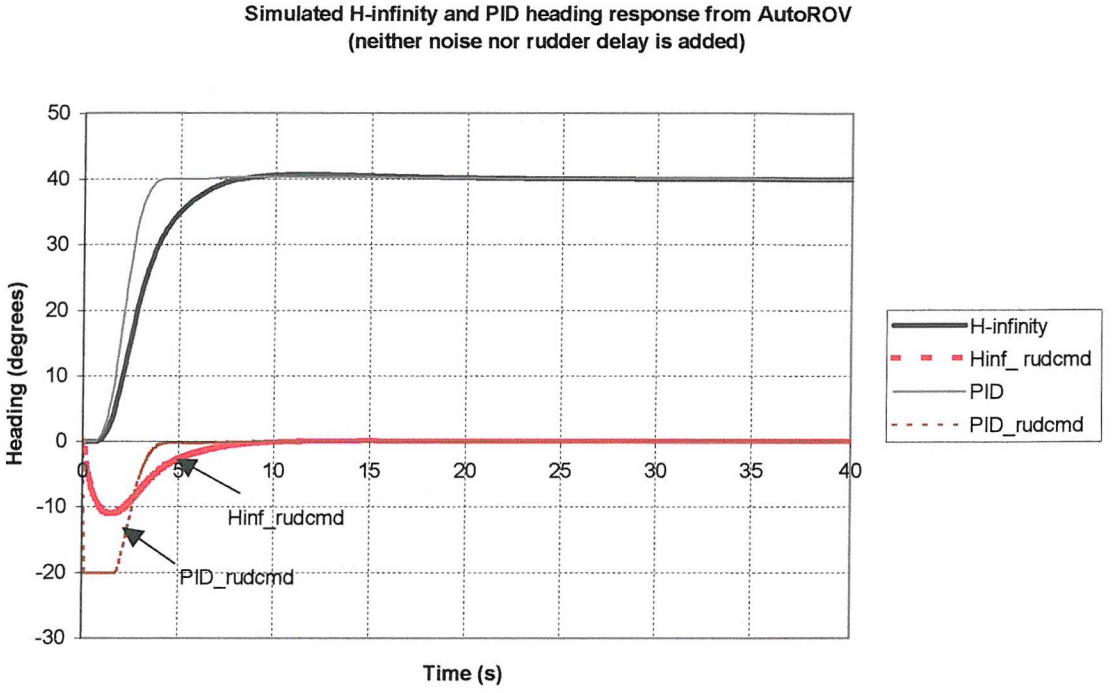


Figure 5.15: Heading responses when neither noise nor rudder delay was added

In figure 5.15 above, the heading response from the PID controller is faster with a rise time of 4 s compared to the H-infinity control with a rise time of 8.5 s. It is noticed that the rudder from the PID control is deflected to its maximum 20 degrees limit and is saturated. The rudder of the underwater vehicle from the H-infinity controller only deflects to about 11 degrees and does not saturate. Thus, there is more control authority on the rudder with the H-infinity control as the output (actual heading angle) tracked the desired heading angle closely.

5.5.3.2 Disturbances, random noise and rudder delay of 0.33s

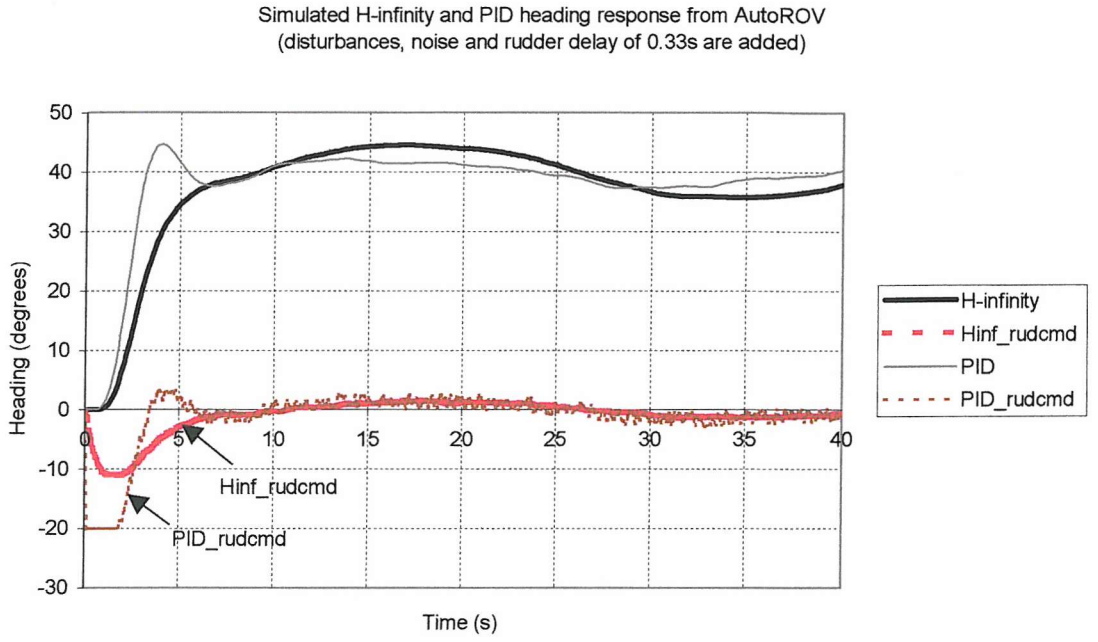


Figure 5.16: Heading response when disturbances, noise and delay of 0.33s are added

Current velocity disturbance, random noise and rudder delay of 0.33s were introduced in this simulation run and its result is shown figure 5.16 above. The PID control produced a ± 2.5 degrees oscillation where as the H-infinity control produced a ± 5 degrees oscillation. The rudder command of the PID control is very noisy compared to the rudder command produced by the underwater vehicle with an H-infinity controller. In addition, the rudder of the PID control saturated at its maximum limit of 20 degrees. However, the rudder of the H-infinity control only deflected to an angle 11 degrees. Because the PID control produced larger rudder deflections, the heading response is faster and caused a larger overshoot of 4.6 degrees compared to the H-infinity controller with no overshoot. In a heading control, an overshoot can have serious implications as it can cause collision with, for example an underwater 'cliff' or an iceberg if under ice.

5.5.3.3 Disturbances, random noise and rudder delay of 0.7s

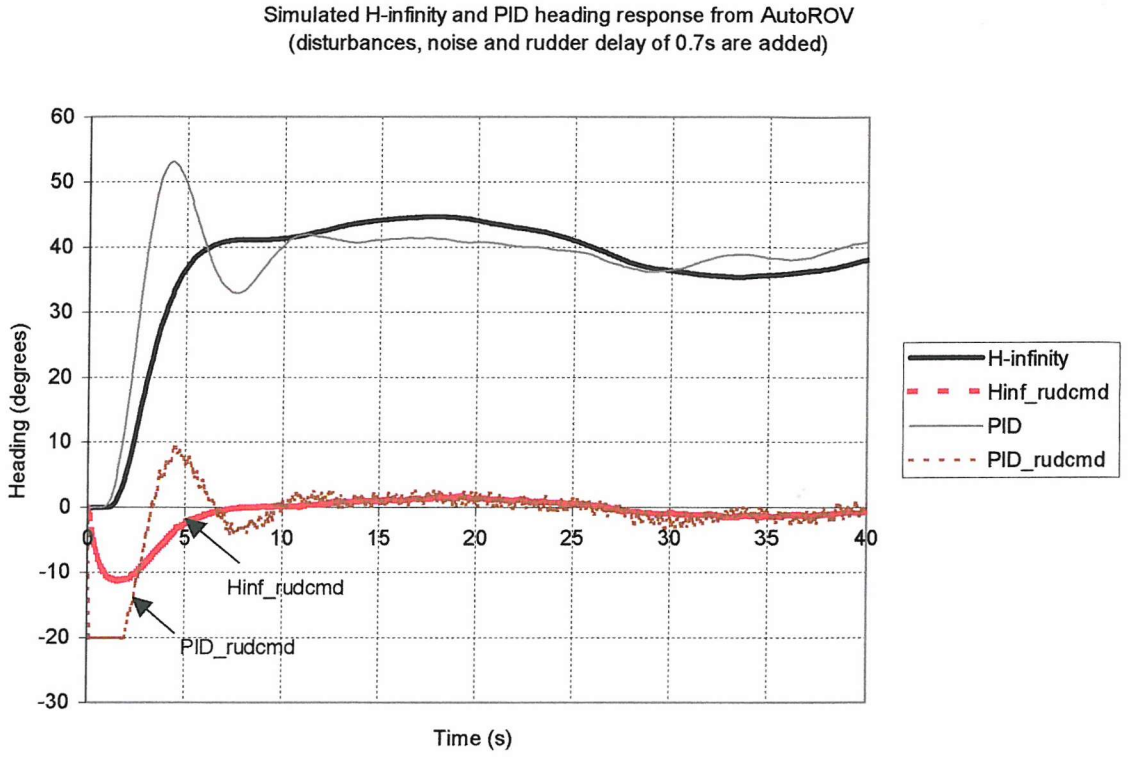


Figure 5.17: Heading response when disturbances, noise and delay of 0.7s are added

This simulation was carried out to obtain the effect of a larger delay in the heading subsystem together with the effect of current velocity and the sensor random noise. Figure 5.17 above shows the heading response of both PID and H-infinity controls. The effects are generally similar to simulation run 5.5.3.2 above but with larger overshoot. This is because the rudder delay introduced was larger than the one presented in figure 5.16. The rudder delay of 0.7s caused the PID control to produce an overshoot of 13 degrees. In comparison, the underwater vehicle with an H-infinity controller produced an overshoot of 1.5 degrees. The PID control produced a noisy and saturated rudder command compared to the H-infinity control. The oscillation caused by the current velocity effects the PID and H-infinity controls by ± 2.5 degrees and ± 5 degrees, respectively.

5.5.3.4 Rudder delay of 0.33s and random noise

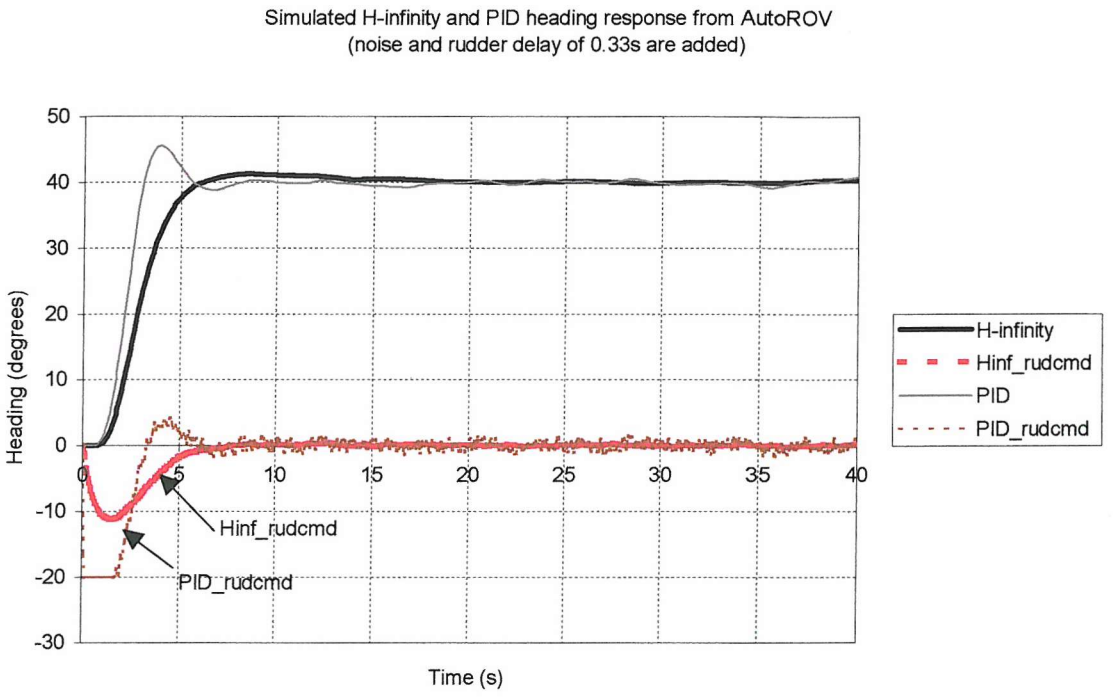


Figure 5.18: Heading responses of PID and H-infinity with 0.33s rudder delay and noise

Random noise and a rudder delay of 0.33s were added during the simulation run. The heading response from the PID control produces a noisy rudder command compared to the rudder command from the H-infinity controller. The rudder deflection operated by the PID control saturated at 20 degrees where as the rudder deflection produced by the H-infinity controller does not have this problem where it only deflects to an angle of 11 degrees. As a result of larger rudder deflection, the PID control produced a faster heading response compared to the H-infinity control. The disadvantage of having a fast response is an overshoot appears in the heading response when the rudder delay presents is large. The overshoot produced by the PID control above is about 5.5 degrees in angle compared to only about 1.1 degrees by the H-infinity controller. Both PID and H-infinity controllers produced similar settling time of 17s.

5.5.3.5 Rudder delay of 0.7s and random noise

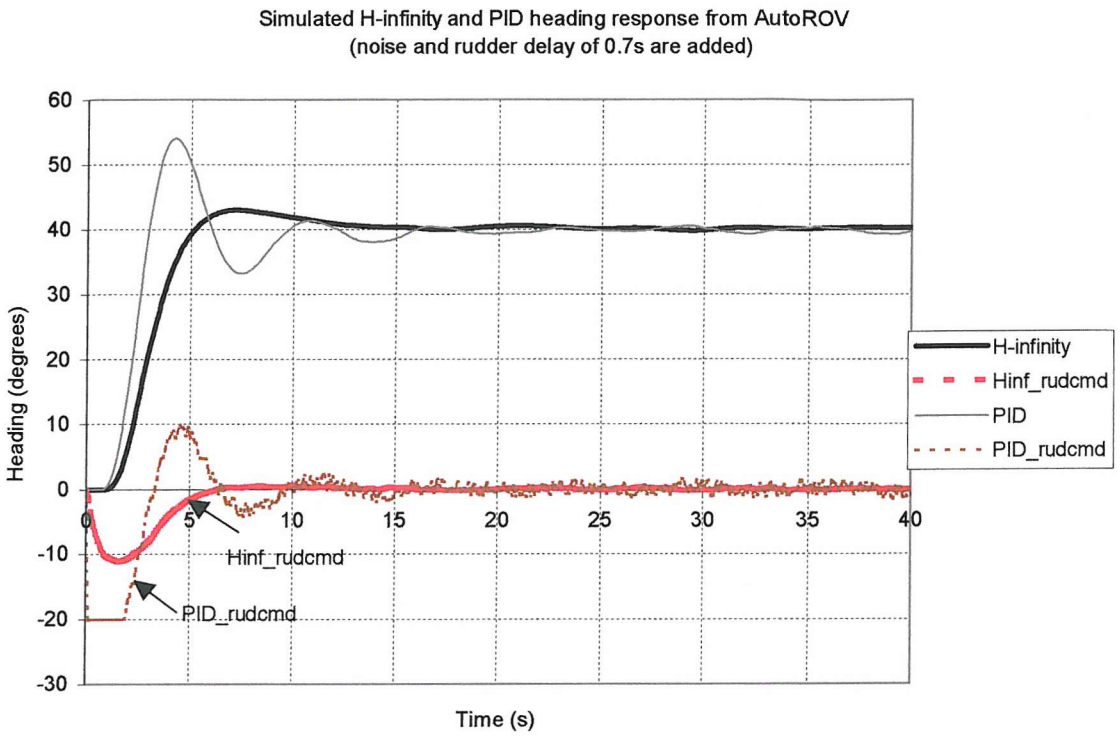


Figure 5.19: Heading responses of PID and H-infinity with 0.7s rudder delay and noise

Figure 5.19 shows the heading responses of simulated PID and H-infinity controllers. A rudder delay of 0.7s and random noise were introduced during the simulation run in *AutoROV*. As the previous simulation result in figure 5.18, the random noise introduced during the simulation run affects the heading response of the PID control and resulted in noisy rudder command as shown in figure 5.19 above. On the other hand, the presence of random noise during the simulation run has less significant effect on the underwater vehicle with the H-infinity controller.

For the PID control, the rudder of the underwater vehicle saturated to 20 degrees, where as the rudder of the underwater vehicle with the H-infinity controller deflects to a maximum of 11 degrees. As a result, the PID produces faster heading response compared to the H-infinity control. However, this resulted an overshoot of 14 degrees and 3 degrees for PID

and H-infinity controls, respectively. Due to a larger overshoot produced by the PID controller, the settling time is ~ 20 s. In comparison, the H-infinity controller results in a faster settling time, compared to PID control, of 17s.

5.5.3.6 Feng Subzero non linear underwater vehicle simulation program

A nonlinear underwater simulation program developed by Feng and Allen [24], was used in the following simulation. The objective was to observe how well the H-infinity controller copes with nonlinear behaviour of an underwater vehicle. The simulation program is programmed using SIMULINK/ MATLAB. It is based on an underwater vehicle, *Subzero II*. The simulation program has no disturbances and no noise added. However, the reading for the heading subsystem only starts after 2 seconds of the simulation. This is to ensure that the vehicle is running constantly at 1.3 ms^{-1}.

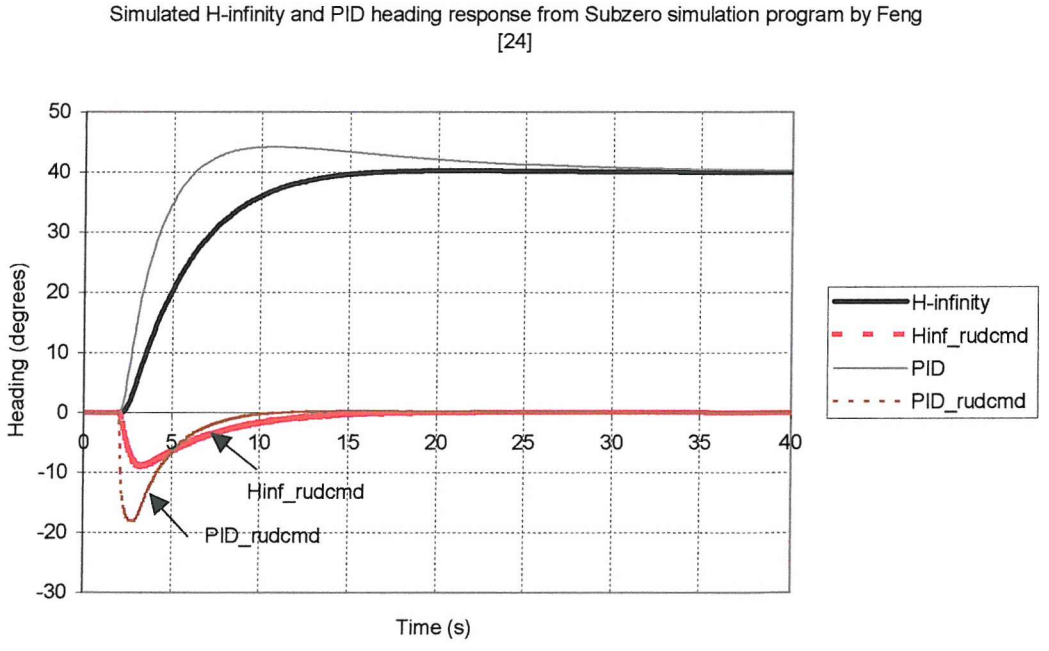


Figure 5.20: Heading responses using a nonlinear underwater vehicle simulation program

The sampling time was set to 0.1s. The layout of the control systems to obtain both H-infinity and PID heading responses are attached in Appendix 8.5

The results of the simulation run are shown in figure 5.20. As can be seen, the PID control causes the rudder of the underwater vehicle to deflect faster than that of the H-infinity control. The rudder deflection of the PID control and the H-infinity control are about 18 degrees and 9 degrees, respectively. As a result, the PID control responded faster than the H-infinity control, but this caused a larger overshoot. The resulting overshoot for the PID control is 4.2 degrees while no overshoot was produced by the H-infinity heading controller. With the PID controller, it takes 30 s before the desired heading angle is attained. On the other hand, it takes only 15 s for the underwater vehicle with the H-infinity controller to settle down to the desired heading angle. The result of each simulation run is summarised in Table 5.2 below.

	PID control	**H-infinity control**
Neither rudder delay nor random noise is added (*5.5.3.1*)	- Faster response - Rudder saturated at 20 degrees - Rise time of 4s	- Rise time of 8.5s - Rudder deflection of 11 degrees
Current velocity disturbance, rudder delay of 0.33s and random noise added (*5.5.3.2*)	- Faster response - Rudder deflection of 20 degrees (saturated) - Overshoot of 4.6 degrees - Oscillation of ± 2.5 degrees	- Rudder deflection of 11 degrees - No overshoot - Oscillation of ± 5 degrees

Current velocity disturbance, rudder delay of 0.7s and random noise added (*5.5.3.3*)	Faster response - Rudder deflection of 20 degrees (saturated) - Overshoot of 13 degrees - Oscillation of ± 2.5 degrees	- Rudder deflection of 11 degrees - Overshoot of 1.5 degrees - Oscillation of ± 5 degrees
Rudder delay of 0.33s and random noise added (*5.5.3.4*)	- Faster response - Rudder deflection of 20 degrees (saturated) - Overshoot of 5.5 degrees - Settling time of 17 s	- Rudder deflection of 11 degrees - Overshoot of 1.1 degrees - Settling time of 17 s
Rudder delay of 0.7s and random noise added (*5.5.3.5*)	- Faster response - Rudder deflection of 20 degrees (saturated) - Overshoot of 14 degrees - Settling time of 20 s	- Rudder deflection of 11 degrees - Overshoot of 3 degrees - Settling time of 17 s
Non linear simulation (*5.5.3.6*)	- Faster response - Rudder deflection of 18 degrees - Overshoot of 4.2 degrees - Settling time of 30 s	- Rudder deflection of 9 degrees - No overshoot - Settling time of 15 s

Table 5.2: Results of the simulation runs

5.5.4 Discussion/ analysis

The results of robust analysis in section 5.4 show that the H-infinity controller fulfilled the closed loop objectives: disturbance rejection, noise attenuation, reference tracking and optimal control of the rudder. The control of the rudder is optimised when the controlling of the rudder is at its minimal level.

From simulations 5.5.3.2 and 5.5.3.3, the graphs show that the underwater vehicle is affected by the sinusoidal water current velocity disturbance. As a result, an oscillatory motion of ± 5 degrees is produced. From figure 5.8, the H-infinity controller should reject any disturbances that may occur. Therefore, the closed loop objective on disturbance rejection is not achieved. This is because the experimental work by Lea [48] was carried out in a tank rather than the sea or a lake. Thus, very little or zero water current velocity is present. Hence, information on water current velocity disturbance is not available in the real data taken by Lea [48]. Furthermore, a water current disturbance model was not included during the design of the H-infinity heading controller. From the applications point of view, discussed in section 1.1.1, most of the AUVs are operated in the sea and at a depth where the effect of water current velocity is small. Water current velocity may only present problems during launching and when the AUV is approaching the surface of the sea after completing a mission. Therefore, it is not crucial for the H-infinity controller to have the ability to reject the large water current velocity, in particular. In addition, from figures 5.16 and 5.17, the sinusoidal water current velocity only caused an oscillation of ± 5 degrees to the heading response of *Subzero II*. The size of the water current oscillation in *AutoROV* is about 4 degrees. The heading response with only random noise and rudder delay added into the simulations can be observed in figure 5.18 and figure 5.19.

Figure 5.9 indicates that the noise is guaranteed to be attenuated by the H-infinity controller. This is proved by the simulation results show in figure 5.16 – figure 5.19. It is seen that the rudder command produced by the underwater vehicle with an H-infinity controller is smaller and less noisy compared to the rudder command produced by the PID control. Therefore, the H-infinity controller is able to attenuate sensor random noise efficiently.

The reference tracking objective is said to be achieved when the underwater vehicle manages to track the desired heading angles closely. From figure 5.15, it can be seen that the heading response of *Subzero II* with an H-infinity controller tracks the desired heading of 40 degrees closely. The time rise produced by the H-infinity control is 8.5 seconds. Thus, the reference tracking closed loop objective is fulfilled.

The optimal control of the rudder is considered to be achieved if only a minimal effort is taken to control the rudder. From figure 5.11, it shows that the H-infinity control fulfilled the optimal control requirement as the singular value of the controller, $K(s)$ is minimised at high frequencies. In the simulation runs above (5.5.3.1 – 5.5.3.5), the maximum rudder deflection produced by the H-infinity control is ~ 11 degrees and resulted in a rise time of 8.5 seconds. In comparison, the PID control managed to produce slightly faster rise time of 4 seconds, but the rudder saturated. Therefore, the H-infinity controller successfully controlled the rudder and avoided saturation.

The requirement of the closed loop transfer function T_{zw} is also satisfied in figure 5.12. Therefore, in general, the H-infinity controller designed is stable as shown in the simulation results in figures 5.15 to 5.19 and from the tabulated results. This is proven when the designed H-infinity controller can withstand random noise and rudder delay well in comparison to the PID control. In addition, the H-infinity controller performed much better compared to the PID control, in the nonlinear underwater simulation as shown in figure 5.20 and from the tabulated results in table 5.2.

From the result listed in Table 5.2, it can be concluded that the PID control produced a faster response in heading control. This is caused by the large rudder deflection which reaches its limit of 20 degrees. The deflection of the rudder is limited by the shaft for the propeller which passes through the rudder linkage. Although the rudder deflection from the H-infinity control produced overshoot, no saturation was noticed. Another advantage from the H-infinity control is that the heading subsystem is insensitive to random noise as the signal of the rudder deflection seems to be *'clean'* from figure 5.16.and figure 5.17.

5.5.1 Possible errors in the design of the H-infinity controller

1) *Discretisation inaccuracy.*

2) *Inaccurate selection of the frequency range for minimising* $S(s)$. This can lead to unnecessary noise amplification and poor stability margins [31].

3) *Inaccurate selection of the other sensitivity functions*: $T(s)$ and $R(s)$. As for $R(s)$, a high pass filter shape might be more an accurate shape to obtain better control on the ruder. In addition, unmodelled dynamics of the linearised heading subsystem and other forms of uncertainties such as modelling error might not have been taken into account during the design of the H-infinity controller.

4) There are some *time delay and lag effects for the control surfaces*. This is reported by Lea [1]. There is difficulty in zeroing the fins of the *Subzero II* and it was found that the accuracy was ± 1 degree. The return zero accuracy was ±2 degrees. The structure of the rudder is such that there are two rudder surfaces and they are linked together to form a single rudder. Due to this kind of structure, there is an offset (with respect to each other) of 5 degrees. This may affect the rudder deflection and thus the heading response.

5.6 Conclusion

From the analysis above, the differences in heading response between the H-infinity control and PID control have been examined. It was found that overshoots were produced when there is delay in the rudder positioning system of the underwater vehicle. However, the overshoot produced by the H-infinity controller is smaller compared to the overshoots produced by the PID control. The underwater vehicle with an H-infinity heading controller is not affected by the random noise. In comparison, fluctuation is seen in the rudder deflection produced by the PID control. Although the H-infinity control produced an oscillatory motion of ± 5 degrees in response to the water current disturbance designed by

Lea [12], it does not affect the whole AUV mission. It might only affect the heading motion during launching and when the AUV is nearing the surface of the sea. Hence, the underwater vehicle with the H-infinity heading controller is more robust compared to the PID heading control when uncertainties such as, rudder delay and random noise are present. From above analysis and investigation, most of the closed loop objectives and other performance specifications listed in section 5.2 above were met.

Performance

From the requirements listed in 5.2.4, most of the specifications have been achieved. From calculations performed below as well as from the simulated heading responses (figure 5.16 and figure 5.17), it was found that the steady state error for the resulting H-infinity heading controller with a step input is zero. This is of course below the desired 0.1 % limit.

From [37], the steady state error, e_∞ is expressed as

$$e_\infty = \lim_{t\to\infty} e(t) = \lim_{t\to\infty} L^{-1}(E(s)) \tag{5.22}$$

, where L^{-1} is the inverse Laplace transform. After transformation, expression (5.22) becomes

$$e_\infty = \lim_{t\to\infty} e(t) = \lim_{s\to 0} sE(s) \tag{5.23}$$

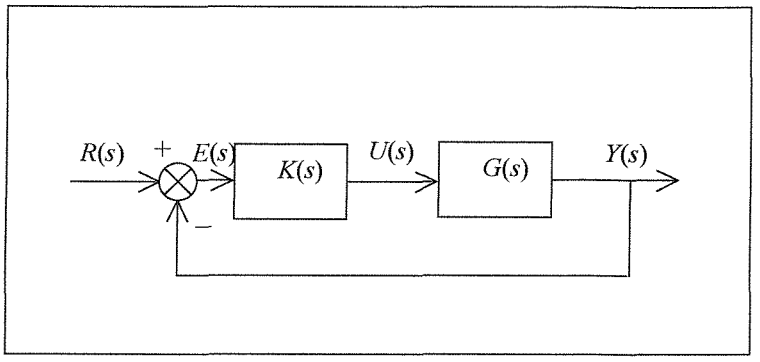


Figure 5.21: A single input – single output (SISO) feedback control system

From figure 5.21, the error signal, $E(s)$ can be expressed as $E(s) = s\dfrac{R(s)}{1+G(s)K(s)}$ where $R(s) = \dfrac{A}{s}$ (the desired input) and A is the magnitude for the step input. Substitute $E(s)$ into expression 5.23, the steady state error becomes

$$e(\infty) = \lim_{s\to 0} s\frac{R(s)}{1+G(s)K(s)}$$

The error signal $E(s)$ was calculated in MATLAB robust control toolbox and it was then substitute into expression 5.23. The resulting steady state is as below.

$$e(\infty) = \lim_{s\to 0} s\frac{(s^{10}+\ldots+2.839e^{8})}{s^{11}+\ldots+1.307e^{8})}A$$

$$e(\infty) = \frac{0}{1.307e^{8}}A, \text{ and } A \text{ is 40 degrees}$$

$$e(\infty) = 0$$

From table 5.2, the overshoots produced by the H-infinity control is smaller compared to the PID control, when uncertainties such as rudder delay are present.

Robustness

From the robust analysis above, all of the closed loop objectives: disturbance rejection, noise attenuation, reference tracking and optimal control for the rudder were met. These are show in figures 5.8 – 5.11. However, the results of the simulated heading response in figures 5.16 and 5.17 do not show that the H-infinity controller is able to reject the current velocity disturbances introduced during the simulation runs. This is because inappropriate data were used to find the bandwidth of the disturbances and a water current disturbance model was not included during the design of the H-infinity heading controller, as discussed in section 5.5.4. The requirement for a stable H-infinity controller is fulfilled where $\| T_{zw} \|_\infty \leq 1$, as shown in figure 5.12. As a result, the H-infinity heading controller managed to withstand the presence of random noise in the system. This is shown by the simulation results in figures 5.15 - 5.19. The H-infinity controller performed very well even with a full nonlinear behaving underwater vehicle simulation as shown in figure 5.20.

Chapter 6

Conclusions to date and Recommendation for further work

6.1 Conclusions to date

6.1.1 H-infinity control

Advantages of using the H-infinity control technique

1) By definition, the H-infinity technique involves minimising the infinity norm of a transfer function, $F(s)$ which is denoted as $\min \| F(s) \|_\infty$. In a single input- single output (SISO) system, the H-infinity norm is simply the maximum value of the frequency response gain. For example, the robust stability for a SISO system is achieved when $\| T(s)W(s) \|_\infty < 1$. Therefore, the robust stability for the SISO system can be obtained by minimising the infinity norm of maximum gain of $T(s)W(s)$ below 1.

2) The objectives and limitations such as minimum steady state error and closed loop bandwidth of a design can be specified through weighting functions $W(s)$ incorporated into the design.

3) Uncertainty models can be included into the H-infinity control design to increase the robustness of the resulting controllers. This is to compensate with errors in modeling as well as external uncertainties such as disturbances and noise.

Some lessons *learned* from using the H-infinity design

1) There is a need to understand fully the theory to effectively design and analyse controllers.

2) The difficult part of the design cycle is to choose the weighting functions. This becomes easier through experience and good understanding of the limitations of the system design.

3) The resulting controller may have a large number of states. This is due to the augmentation of the model and weighting functions during the design stage. Therefore, the controller needs good model reduction techniques to reduce the order.

4) It is impractical to force the singular value, σ below unity for all frequencies, by tuning the weighting functions. Thus, it can be a tedious iteration process.

6.1.2 Robust analysis

Several analyses were carried out using the MATLAB robust control toolbox on the robustness of the H-infinity controller designed. There are several closed loop objectives to be fulfilled in order to obtain a robust system. They are disturbance rejection, noise attenuation, reference tracking and optimal control of the actuator. From the analyses performed, all of the closed loop objectives are successfully achieved.

6.1.3 Simulated heading response

The heading responses obtained using the H-infinity controller were quite similar to those obtained using the PID controller. The PID controller was designed by Lea [1]. With the presence of uncertainties such as rudder delay and noise, however, the underwater vehicle with the PID controller was less stable.

1) The rise time produced by the H-infinity controller was slower than the rise time produced by the PID controller. No saturation is observed on the rudder command produced by the H-infinity control, while the rudder command of the PID control experienced a saturation problem. The saturation problem experienced with PID control does not happen in the simulation package designed by Feng [47]. Therefore, the anti-windup integrator used in the heading subsystem in *AutoROV* by Lea [1], failed to prevent the saturation problem for the PID heading controller.

2) The H-infinity control withstood the presence of random noise introduced during the simulation runs in *AutoROV* very well in comparison to the PID control.

3) The size of overshoot from the H-infinity control was smaller compared to the overshoots produced by the underwater vehicle with the PID controller.

4) The sinusoidal water current velocity introduced during the simulation caused an oscillatory motion with both controllers. A water current disturbance model was not included during the design of the H-infinity heading controller. Therefore, the H-infinity heading controller was not expected to withstand the water current disturbance added in the simulation runs 5.5.3.2 and 5.5.3.3. In addition, the experimental data used to obtain the bandwidth of disturbance for the H-infinity control is not appropriate. This is because the underwater vehicle was tested in a tank which have practically zero or very little water current velocity.

5) The underwater vehicle with the H-infinity control copes very well with modelled nonlinear behaviour, compared to the PID control.

In general, the H-infinity control is more stable and robust. A summary of the heading performance of both H-infinity and PID controllers are tabulated below.

H-infinity control	PID control
1) Insensitive to random noise 2) Lower rudder deflection produced lower overshoot	1) Fast response but as a result, an overshoot is produced

Table 6.1: Summary of the simulated heading performance and robustness for the H-infinity and the PID controls

6.2 Further work

From the above discussion, several areas of further work can be suggested :-

1) To carry out several experimental tests with the resulting H-infinity controller (5.15) using the underwater vehicle, *Subzero II*. The H-infinity heading controller was designed based on data obtained from heading control experimental work but the tests were carried out in tank. Therefore, it is preferable that future heading control tests are carried out in tank rather than, for example, in a lake. This would facilitate validation of the simulated heading responses in simulation runs 5.5.3.4 and 5.5.3.5, without the presence of water current velocity disturbances. It is expected that the experimental responses will be close to the simulated results.

2) To obtain the bandwidth of the water current velocity disturbance from real data. This information is then used in the performance weighting function, $W_1(s)$ for designing an H-infinity controller to ensure that the controller can withstand any such disturbances present.

3) To obtain information on the yaw disturbance from real data and to include this information in the control weighting function, $W_2(s)$. For course-keeping, it is required that the modified control weighting function has a high pass filter shape [10]. This is to have a high magnitude of the control sensitivity function, $R(s)$ at the low frequency for maximum rudder control action and a low magnitude of the control sensitivity function at the high frequency for minimum rudder control action, so that $R(s) < W_2(s)^{-1}$. A maximum rudder control action at the low frequencies is needed to attenuate the yaw disturbances while a minimum rudder control action at the high frequencies is required to prevent saturation.

4) To increase the robustness of the heading control subsystem by including a model of uncertainties such as noise and external wave disturbance, in the design of the H-infinity control. If an uncertainty is added in the plant, $G(s)$, another additional two closed loop objectives can be determined [31]. There are:

a) Robust stability when there is an additive uncertainty: $\bar{\sigma}(R(s))$ small or minimised $\bar{\sigma}(K)$

b) Robust stability when there is a multiplicative uncertainty: $\bar{\sigma}(T(s))$ small or minimised $\bar{\sigma}(GK)$

A brief description on uncertainty can be found in Appendix 8.7. Detailed explanation on uncertainty is discussed in [31].

5) Consider model reduction and ordered balanced realisation for H-infinity controllers so as to maximise the performance [43]. The order of the current H-infinity controller, shown in expression (5.15) may be high. This is due to the augmentation process between the weighting function and the plant dynamics. Therefore, it is advisable to keep the order of the weighting functions as low as possible, but it is sometimes difficult because of the complex nature of the system and its environment. Alternatively, a *model reduction method* is used to lower the order of the controller designed. With *ordered balanced*

realisation, it ensures that the controller and the system are both controllable and observable. This is done by eliminating the unwanted states. In addition, it is important to make sure that the reduced-order model is similar or almost similar to the original model. A brief explanation and some examples of model reduction as well as order balanced realization can be obtained in [43].

6) To design H-infinity controllers for the depth and speed subsystems, based on real data obtained by Lea [48]. Then, all three subsystems: speed, heading and depth, should be coupled as a multiple input – multiple output (MIMO) system.

7 References

AUV

[1] Lea, R K, Allen, R, Merry, S L, A Comparative Study by Simulation and Experimentation of Control Techniques for Autonomous Underwater Flight Vehicle, PhD Thesis Institute of Sound and Vibration Research (ISVR), University of Southampton, May 1998.

[2] McPhail, S D and Pebody, M, Autosub-1. A Distributed Approach to Navigation and Control of an Autonomous underwater Vehicle. (www.soc.soton.ac.uk/OTD/asub/papers/iee97p3.html)

[3] Short articles on Autosub-1 (www.soc.soton.ac.uk/OTD/asub/)

[4] Liceaga-Castro, E and Molen, G van der., A Submarine H_∞ Depth Control Under Wave Disturbances, *IEEE Transactions on Control Systems Technology*, v3, n3, pp 338-346 (1995).

[5] Jalving, B and Storkersen, N, The Control system of an autonomous underwater vehicle, *Modeling, Identification and Control*, v16, n2, pp 107-117, 1995; Jalving B, The NDRE-AUV Flight Control System, *IEEE Journal of Oceanic Engineering*, v19, n4, pp497-501, October 1994.

[6] Healey A J and Lienard D, Multivariable Sliding Mode Control for autonomous Diving and steering of unmanned underwater vehicles, *IEEE Journal of Oceanic Engineering*, v18, n3, July 1993.

[7] Silvestre C and Pascoal A, Control of an AUV in the vertical and horizontal planes: system design and tests at sea, *Transaction of Institute of Measurement and Control*, v19, n3, 1997.

[8] Seube, N, Review of Control Methods for Underwater Vehicles Navigation in Uncertain Environment, *Oceans '94: Oceans engineering for today's technology and tomorrows preservation. Proceedings Vol III*, pp99-104, 1994.

[9] Logan, C L, A Comparison between H-Infinity /Mu-Synthesis Control and Sliding Mode Control for Robust Control of a Small Autonomous Underwater Vehicle, *Proceedings of IEEE Symposium on Autonomous Underwater vehicle Technology (AUV 1994)*, New York 1994.

[10] Agarwal, A, H_∞ robust control technology applied to the design of a combined steering/stabliser systems for warships, *11th Ship Control Systems Symposium*, v1 (1997).

[11] Cowling, D, Entire Operating Envelope H-Infinity Design for An Unmanned Underwater Vehicle, *Oceans 96 MTS/IEEE: The Coastal Ocean-Prospects for the 21st Century* 23-26 September 1996, Florida, pp 24-29, 1996.

[12] Lea, R K, *AUTOROV- An Underwater Flight Vehicle Simulation Program*, Institute of Sound and Vibration Research (ISVR) Technical Memorandum no. 828, July 1998.

[13] Schexnayder K R, Pettway W C, Sharp K M and Lammons G, New Generation AUVs Enter Navy Operations Arena (Lighter, Smaller Vehicles Demonstrate Commercial Capabilities, *Sea Technology Magazine*, v41, n12, December 2000.

[14] Kojima J, Kato Y, Asakawa K, Matumoto S, Takagi S, Kato N, Development of Autonomous Underwater Vehicle 'Aqua Explorer 2' for Inspection of Underwater Cables, *IEEE OCEANS' 1997.*

[15] Laval B, Bird J S and Helland P D, An Autonomous Underwater Vehicle for the Study of Small Lakes, *Journal of Atmospheric and Oceanic Technology*, v17, pp 69-76, January 2000.

[16] White D G, Autonomous Undersea Vehicles Part 1 – The Test Bed AUVs, *Ocean News and Technology Magazine July/August 1999*, pp 30-32.

[17] White D G, Autonomous Undersea Vehicles Part 2 – AUVs for Search and Survey, *Ocean News and Technology Magazine September/October 1999*, pp 30-32.

[18] Wernli R L, AUV Commercialization – Who's Leading the Pack?, *OCEANS 2000 MTS/ IEEE Conference Proceedings 2*, Providence, RI, 11-14 September 2000.

[19] Wernli R L, AUV's – The Maturity of the Technology, *OCEANS '99 MTS/ IEEE Conference Proceedings 2*, Seattle, WA, 13-16 September 1999.

[20] Gaskett C, Wettergreen D and Zelinsky A, Reinforcement Learning applied to the control of an Autonomous Underwater Vehicle, *Proceedings of the Australian Conference on Robotics and Automations (AUCRA '99)*, 1999.

[21] Antonelli G, Chiaverini S, Sarkar N and West M, Adaptive Control of an Autonomous Underwater Vehicle: Experimental Results on ODIN, *IEEE Transactions on Control Systems Technology*, v9, n5, September 2001.

[22] Hyde, R A, Glover, K and Shanks, G T, VSTOVL first flight of an H-infinity control law, *Computing and Control Engineering Journal (IEE)*, v6, n1, Feb. 1995.

[23] Venugopal K P, Sudhakar R and Pandya A S, On-line learning control of autonomous underwater vehicles using feedforward neural networks, *IEEE Journal of Oceanic Engineering*, v17, n4, pp308-319, Oct. 1992.

[24] Feng Z and Allen R, H-infinity autopilot design for an autonomous underwater vehicle, *Proceedings of the 2002 IEEE International Conference on Control Applications and International Symposium on Computer Aided Control Systems Design*, September 18-20, 2002, Glasgow, Scotland, U.K. PP 350-354

Simulator and vehicle dynamics

[25] Lauvdal T and Fossen T I, MATLAB Simulator, 1995, www.goggle.com

[26] J. Yuh, Modeling and Control of Underwater Robotic Vehicles, *IEEE Transactions on System, Man and Cybernetics*, V20, n6, pp 1475-1483, Nov/Dec 1990.

[27] Fossen, T I, *Guidance and Control of Ocean Vehicles*, John Wiley & Sons, 1994.

[28] Stanford A L, Tanner J M, *Physics for Students of Science and Engineering*, Academic P., April1985.

Theory

[29] Shinners, S M, *Modern Control System Theory and Design (Second Edition)*, John Wiley & Sons, Inc., A Wiley-Interscience Publication, 1998.

[30] Dutton K, Thompson S, and Barraclough, B, *The Art of Control Engineering*, Addison Wesley Longman, April 1997.

[31] Skogestad, S. and Postlethwaite, I. (1996), *Multivariable Feedback Control*, John Wiley & Sons, Chichester, UK.

[32] Helton, J. W. and Merino, O., *Classical Control Using H∞ Methods (An Introduction to Design)*, SIAM Society for Industrial & Applied Mathematics, August 1998.

[33] Francis, B A, *A Course in H-infinity Control Theory*, Lecture Notes in Control and Information Sciences, v 88, Springer-Verlag, 1987.

[34] Kwakernaak, H, Robust Control and H∞ - optimization - Tutorial Paper, *Automatica*, v29, n2, pp 255-273, 1993.

[35] Grimble, M J and Johnson, M A (Industrial Control Unit, University of Strathclde), H_∞ Robust Control Design – A Tutorial Review, *Computing & Control Engineering Journal*, November 1991.

[36] Chiang R Y and Safonov M G, H-infinity synthesis Using a Bilinear Pole Shifting Transform, *Journal of Guidance, Control and Dynamics*, V15 (5), pg 1111-1117, 1992

[37] Tewari A, *Modern Control Design with MATLAB and SIMULINK*, John Wiley & Sons Ltd, 2002

[38] Glyn James *et al*, Advanced Modern Engineering Mathematics, Addison Wesley, Addison- Wesley Publishing Company, 1993

[39] Kapsenberg G K, "A step towards the introduction of simulation techniques in the world of remotely operated vehicles", *International Shipbuilding Progress*, vol 32 pp 92-98, 1985.

[40] Kalske, S, "Motion simulation of underwater vehicles", Tech. Rep. VVT-PUBS-97, Technical Research Centre of Finland (VVT), April 1992.

[41] Mahieu F, Folleco A, An E, Smith S, Software Development of the Hardware-in-the-loop Implementation and 3D Viewer for Ocean Explorer, *SPIE 2000*, April, Orlando, Florida.

[42] Doucy O, Brutzman D and Healey A, Near Surface Manoeuvring and Station-Keeping for an Autonomous underwater vehicle, presented at *NATO Symposium, Applied Vehicle Technology Panel*, Ankara, Turkey, October 2000-08-29.

[43] Chiang R Y and Safonov M G, *Robust Control Toolbox For Use with MATLAB User's Guide version 2*, The Mathwork Inc., 1999 .

[44] Gates P J and Lynn N M, *Ships, Submarines and the sea: Volume two*, Brassey's Sea power: Naval Vessels, Weapons Systems and technology series, Brassey's (UK) 1990.

[45] Vaccaro R J, *Digital control: A State-Space Approach*, McGraw-Hill International Editions, Electrical Engineering Series, Singapore, 1995.

[46] Lyons, R G, *Understanding Digital Signal Processing*, Prentice Hall PTR, 2001

[47] Feng Z and Allen R, *Modelling of Subzero II*, Technical Memorandum No 880, ISVR, University of Southampton, October 2001

[48] Lea R K, Haslar Tank Test notes, April 1997 – September 1997.

[49] Brost B, Johnson B and Tulipani D, Underwater Noise Pollution and Marine Mammals, *MSCI 375, Biology of Marine Mammals, 1998.*

Bibliography

AUV

Desanj, D. S, Donha, D. C, Katebi, M. R and Grimble, M. J, H_∞ adaptive controllers for auto-pilot applications, *11th Ship Control Systems Symposium*, v1 (1997).

Hyde, R.A, Glover, K. and Shanks, G. T, VSTOL First Flight of an H_∞ Control Law, *Computing & Control Engineering Journal*, v6, n1, November 1991.

The Mathworks Home, MATLAB Digest, www.mathworks.com/company/digest/march00/autopilot/

Short article from FAU (www.oe.fau.edu/AMS/auv.html)

Articles from the *Sea Technology* magazine

Nomenclature for treating the motion of a submerged body through a fluid, Technical and Research Bulletin 1-5, The Society of Naval Architects and Marine Engineers, Oct. 1964.

Conte G and Serrani A, Robust Control of a Remotely Operated Underwater Vehicle, *Automatica*, v 34, n2, pp 193-198, 1998.

Kaminer I *et al*, Control of an Underwater Vehicle using H_∞ Synthesis, *Proceedings of the 30th Conference on Decision and Control*, Brighton, England, December 1991.

Skjetne R, Case Study: Controlling an Underwater ROV with a Robust Linear Control System, *ECE 232 – Robust Control Systems*, Spring 1999.

Theory

Sanchez-Pena, R, Conea (Buenos Aires, Argentina) and Sznaier, M. (Pennysylvania State University, Philadelphia, USA), *Robust Systems Theory and Applications*, John Wiley & Sons, USA (1998).

Chen, B. M, H-infinity Control and Its Applications, Lecture Notes in Control and Information Sciences, v 235, Springer, 1998.

Zhou, K., Doyle, J. C., Glover, K., *Robust and Optimal Control*, Prentice Hall, New Jersey, USA 1996.

Short articles from **www.goggle.com**

8 Appendix

8.1 Equation of motions (Six Degrees of Freedom)

The nonlinear equation of motions for an underwater vehicle is given below (8.1) – (8.6) as in Lea [1]. The definition of the variables are listed in Table A.

The equation of motion for surge:

$$\left(m-\frac{\rho}{2}l^3X'_{\dot{u}}\right)\dot{u}+mz_G\dot{q}-my_G\dot{r}=\frac{\rho}{2}l^4\left[X'_{qq}q^2+X'_{rr}r^2+X'_{rp}rp\right]$$
$$+\frac{\rho}{2}l^3\left[X'_{vr}vr+X'_{wq}wq\right]$$
$$+\frac{\rho}{2}l^2\left[X'_{vv}v^2+X'_{ww}w^2\right]$$
$$+\frac{\rho}{2}l^2\left[X'_{\delta r\delta r}\delta r^2+X'_{\delta s\delta s}\delta s^2\right]u^2-(W-B)\sin\theta$$
$$+F_{prop}-\frac{\rho}{2}l^2X'_{uu}u^2$$
$$+m\left[vr-wq+x_G(q^2+r^2)-y_Gqp-z_Grp\right] \quad \textbf{(8.1)}$$

The equation of motion for sway:

$$\left(m-\frac{\rho}{2}l^3Y'_{\dot{v}}\right)\dot{v}-\left(mz_G+\frac{\rho}{2}l^4Y'_{\dot{p}}\right)\dot{p}+\left(mx_G-\frac{\rho}{2}l^4Y'_{\dot{r}}\right)\dot{r}=\frac{\rho}{2}l^4\left[Y'_{pp}p\mid p\mid+Y'_{pq}pq\right]$$
$$+\frac{\rho}{2}l^3\left[Y'_{r}ur+Y'_{p}up+Y'_{wp}wp\right]$$
$$+\frac{\rho}{2}l^2\left[Y'_{uu}u^2+Y'_{v}uv+Y'_{vv}v\sqrt{v^2+w^2}\right]$$
$$+\frac{\rho}{2}l^2Y'_{\delta r}u^2\delta r+(W-B)\cos\theta\sin\phi$$
$$-\frac{\rho}{2}C_d\int_{X_{tail}}^{X_{nose}}y(x)(v+xr)\sqrt{(w-xq)^2+(v+xr)^2}dx$$
$$+m\left[wp-ur+y_G(r^2+p^2)-z_Gqr-x_Gqp\right]$$

(8.2)

The equation of motion for heave:

$$\left(m-\frac{\rho}{2}l^3Z'_{\dot{w}}\right)\dot{w}+my_G\dot{p}-\left(mx_G+\frac{\rho}{2}l^4Z'_{\dot{q}}\right)\dot{q}=\frac{\rho}{2}l^3\left[Z'_{q}uq+Z'_{vp}vp\right]$$
$$+\frac{\rho}{2}l^2\left[Z'_{uu}u^2+Z'_{w}uw\right]$$
$$+\frac{\rho}{2}l^2\left[Z'_{w^*}u\mid w\mid+Z'_{ww}\mid w\sqrt{v^2+w^2}\right]$$
$$+\frac{\rho}{2}l^2Z'_{\delta s}u^2\delta s+(W-B)\cos\theta\cos\phi$$
$$-\frac{\rho}{2}C_d\int_{X_{tail}}^{X_{nose}}y(x)(w-xq)\sqrt{(w-xq)^2+(v+xr)^2}dx$$
$$+m\left[uq-vp+z_G(p^2+q^2)-x_Grp-y_Grq\right]$$

(8.3)

The equation of motion for roll

$$-\left(mz_G+\frac{\rho}{2}l^4K'_{\dot v}\right)\dot v+my_G\dot w+\left(I_x-\frac{\rho}{2}l^5K'_{\dot p}\right)\dot p-I_{xy}\dot q-\left(I_{zx}+\frac{\rho}{2}l^5K'_{\dot r}\right)\dot r$$
$$=\frac{\rho}{2}l^5\left[K'_{qr}qr+K'_{pp}p|p|\right]+\frac{\rho}{2}l^4\left[K'_{p}up+K'_{r}ur+K'_{wp}wp\right]$$
$$+\frac{\rho}{2}l^3\left[K'_{uu}u^2+K'_{vR}uv\right]$$
$$+\frac{\rho}{2}l^3K'_{\delta r}u^2\delta r$$
$$+(y_GW-y_BB)\cos\theta\sin\phi+\frac{\rho}{2}l^3K'_{prop}n^2$$
$$-(I_z-I_y)qr+I_{zx}qp-(r^2-q^2)I_{yz}-I_{xy}pr$$
$$+m\left[y_G(uq-vp)-z_G(wp+ur)\right]$$

(8.4)

The equation of motion for pitch

$$mz_G\dot u-\left(mx_G+\frac{\rho}{2}l^4M'_{\dot w}\right)\dot w-I_{xy}\dot p+\left(I_y-\frac{\rho}{2}l^5M'_{\dot q}\right)\dot q-I_{yz}\dot r$$
$$=\frac{\rho}{2}l^5M'_{rp}rp+\frac{\rho}{2}l^4M'_{q}uq$$
$$+\frac{\rho}{2}l^3\left[M'_{uu}u^2+M'_{w}uw+M'_{wwR}w\sqrt{v^2+w^2}\right]$$
$$+\frac{\rho}{2}l^3\left[M'_{w*}u|w|+M'_{ww}|w\sqrt{v^2+w^2}|\right]+\frac{\rho}{2}l^3M'_{\delta s}u^2\delta s$$
$$+\frac{\rho}{2}C_d\int_{X_{tail}}^{X_{nose}}y(x)(w-xq)\sqrt{(w-xq)^2+(v+xr)^2}xdx$$
$$-(x_GW-x_BB)\cos\theta\cos\phi-(z_GW-z_BB)\sin\theta$$
$$-(I_x-I_z)rp+I_{xy}qr-(p^2-r^2)I_{zx}-I_{yz}qp$$
$$-m\left[z_G(wq-vr)+x_G(uq-vp)\right]$$

(8.5)

The equation of motion for yaw:

$$
\begin{aligned}
-my_G\dot{u} + \left(mx_G - \frac{\rho}{2}l^4 N'_{\dot{v}}\right)\dot{v} - \left(I_{zx} + \frac{\rho}{2}l^5 N'_{\dot{p}}\right)\dot{p} - I_{yz}\dot{q} + \left(I_z - \frac{\rho}{2}l^5 N'_{\dot{r}}\right)\dot{r} \\
= \frac{\rho}{2}l^5 N'_{pq}pq + \frac{\rho}{2}l^4\left[N'_p up + N'_r ur\right] \\
+ \frac{\rho}{2}l^3\left[N'_{uu}u^2 + N'_v uv + N'_{vvR}v\sqrt{v^2+w^2}\right] \\
+ \frac{\rho}{2}l^3 N'_{\delta r}u^2\delta r \\
- \frac{\rho}{2}C_d\int_{X_{tail}}^{X_{nose}} y(x)(v+xr)\sqrt{(w-xq)^2+(v+xr)^2}\,xdx \\
-(x_G W - x_B B)\cos\theta\sin\phi + (y_G W - y_B B)\sin\theta \\
-(I_y - I_x)pq + I_{yz}rp - (q^2 - p^2)I_{xy} - I_{zx}rq \\
+ m\left[x_G(wp - ur) - y_G(vr - wq)\right]
\end{aligned}
$$

(8.6)

List of variables

m	Mass of the vehicle (kg)
l	Length of the vehicle (m)
ρ	Water density (kgm^{-3}) ~ 1000 kgm^{-3}
x_G, y_G, z_G	Position of the vehicle's center of mass
u	Velocity of the vehicle in the local x-axis or surge speed (ms^{-1})
v	Velocity of the vehicle in the local y-axis or sway speed (ms^{-1})
w	Velocity of the vehicle in the local z-axis or heave speed (ms^{-1})
p	Roll rate ($rads^{-1}$)
q	Pitch rate ($rads^{-1}$)
r	Yaw rate ($rads^{-1}$)
δr	Rudder deflection (rad)
δs	Sternplane deflection (rad)

W	Weight of the vehicle (N)
B	Buoyancy of the vehicle (N)
F_{prop}	Thrust produced by the propeller (N)
ϕ	Vehicle global roll angle (rad)
ψ	Vehicle global yaw angle (rad)
θ	Vehicle global pitch angle (rad)
X, Y, Z	Forces in the local x-, y-, z-axes of the vehicle, respectively
K, M, N	Moment about the local x-, y-, z-axes of the vehicle, respectively
I_x, I_y, I_z	Moment of inertia about the local x-, y-, z-axes of the vehicle, respectively
C_d	Sideways drag coefficient of the vehicle
$y(x)$	Diameter of the vehicle at a distance x from tail to nose
X_{pq}	An example of a non dimensional hydrodynamics coefficient. It represents the partial derivative of the hydrodynamic force in the X direction with respect to the roll rate (p) and the pitch rate (q).

Table A: List of variables for the nonlinear equation of motions

8.2 Solutions to H-infinity control: State Space solution /Riccati Equation solution

The system is assumed to be stable and $G(s)$ has the state space realisation

$\begin{pmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & 0 \end{pmatrix}$. A matrix called the Hamiltonion matrix is used to obtain the solutions to the Riccati Equation from its eigenvalues and eigenvectors. The Riccati equation is a

first order, nonlinear differential equation and it is solved by numerical methods. The solution is a matrix rather than a vector. Further explanation on the Riccati equation can be found in [30], [31].

The Hamiltonion matrix is denoted as **H** and is represented by:

$$\mathbf{H} = \begin{pmatrix} A & -BB^T \\ -C^TC & -A^T \end{pmatrix}.$$

Elements A, B and C are taken from state space equivalent of $G(s)$ above. If there are no eigenvalues, λ on the imaginary axis then the condition is stable and $P = X_2X_1^{-1}$. X_1 and X_2 are the eigenvectors and X_1 must be a square matrix. The corresponding *Riccati equation* is $A^TP + PA - P(BB^T)P + C^TC = 0$. In the case of H-infinity control law, the above state space $G(s)$ is replaced by the augmented plant, $P(s)$. The matrix $P(s)$ has the form of

$$P = \begin{pmatrix} A & B_1 & B_2 \\ C_1 & D_{11} & D_{12} \\ C_2 & D_{21} & D_{22} \end{pmatrix}$$

To calculate the infinity norm, the Hamiltonion matrix, **H** is employed to approximate the solution to multivariable systems. (If **H** does not have any imaginary eigenvalues, then $\| \mathbf{G} \|_\infty < \gamma$). For the H-infinity design problem, two Hamiltonion matrices are used: -

$$\mathbf{H}_{X\infty} = \begin{pmatrix} A & \gamma^{-2}B_1B_1^T - B_2B_2^T \\ -C_1^TC_1 & -A^T \end{pmatrix} \quad ($$

$$\mathbf{H}_{Y\infty} = \begin{pmatrix} A^T & \gamma^{-2} C_1{}^T C_1 - C_2{}^T C_2 \\ -B_1 B_1{}^T & -A \end{pmatrix}$$

(8.7)

As mentioned above, the Hamiltonion matices above are used to find the solutions for the Riccati equations. X_∞ and Y_∞ are the combined eigenvectors of $\mathbf{H}_{X\infty}$ and $\mathbf{H}_{Y\infty}$. A stability condition exists if there are no eigenvalues on the imaginary axis such that $X_\infty = X_2 X_1^{-1}$. X_1 and X_2 are the eigenvectors and X_1 must be a square matrix. The corresponding Riccati equations for the above matices are

$$A^{\mathrm{T}} X_\infty + X_\infty A + X_\infty (\gamma^{-2} B_1 B_1{}^{\mathrm{T}} - B_2 B_2{}^{\mathrm{T}}) X_\infty + C_1{}^{\mathrm{T}} C_1 = 0$$

(8.8)

$$A Y_\infty + Y_\infty A^{\mathrm{T}} + Y_\infty (\gamma^{-2} C_1{}^{\mathrm{T}} C_1 - C_2{}^{\mathrm{T}} C_2) Y_\infty + B_1 B_1{}^{\mathrm{T}} = 0$$

(8.9)

In practice, it is often sufficient to have an H_∞ sub-optimal controller. With this, let $\gamma_{\min}$ be the minimum value of the transfer function over all stabilising controllers $K(s)$, such that $\| T_{zw} \|_\infty \leq \gamma$ if and only if the three conditions are met simultaneously. This is solved by reducing γ, iteratively. There exists a stabilising controller if and only if the three conditions below are met simultaneously:

a) $X_\infty \geq 0$ is a solution to (8.8)

$\Re\, \lambda_i [A + (\gamma^{-2} B_1 B_1{}^{\mathrm{T}} - B_2 B_2{}^{\mathrm{T}}) X_\infty] < 0$; for all the real ($\Re$) part of i^{th} eigenvalue λ

(8.10)

b) $Y_\infty \geq 0$ is a solution to (8.9)

$$\Re\, \lambda_i [A + Y_\infty (\gamma^{-2} C_1^{\mathrm{T}} C_1 - C_2^{\mathrm{T}} C_2)] < 0; \text{ for all the real } (\Re) \text{ part of } i^{\text{th}} \text{ eigenvalue } \lambda \tag{8.11}$$

c) $\rho(X_\infty Y_\infty) < \gamma^2$, where ρ is the spectral radius or the largest eigenvalue

(8.12)

From this the controller gain can be represented as

$$\mathbf{K} = \begin{pmatrix} A + \gamma^{-2} B_1 B_2^{\mathrm{T}} X_\infty + B_2 C_k - B_k C_2 & B_k \\ C_k & 0 \end{pmatrix} \tag{8.13}$$

where $B_k = (I - \gamma^{-2} Y_\infty X_\infty)^{-1} (Y_\infty C_2^{\mathrm{T}})$ and $C_k = -B_2^{\mathrm{T}} X_\infty$.

In [31], it is represented as

$$\mathbf{K} = \begin{pmatrix} A + \gamma^{-2} B_1 B_2^{\mathrm{T}} X_\infty + B_2 C_k - B_k C_2 & (I - \gamma^{-2} Y_\infty X_\infty)^{-1} (Y_\infty C_2^{\mathrm{T}}) & (I - \gamma^{-2} Y_\infty X_\infty)^{-1} B_2 \\ -B_2^{\mathrm{T}} X_\infty & 0 & I \\ C_2 & I & 0 \end{pmatrix}$$

The summary for above solution:-

1) The augmented plant $P(s)$ from the plant dynamics $G(s)$ and weighting functions $W(s)$ are obtained.
2) Matrices (8.7) are obtained.
3) Their respective eigenvalues and eigenvectors are found.
4) From the eigenvectors, the combined eigenvectors X_∞ and Y_∞ are computed.
5) The solutions to X_∞ and Y_∞ are validated by (8.8) and (8.9).
6) A stabilising H-infinity controller is obtained by satisfying requirements (8.10), (8.11) and (8.12).

Procedure to obtain a stabilising H-infinity controller:-

1) The controller $K(s)$ is found by using (8.13). For the H_2 problem, $\gamma = \infty$.
2) $\| T_{zw} \|_\infty$ is computed which in turn, gives the γ_{upper} bound.
3) The γ-iteration process is performed.
4) The γ_{opt} is substituted to $K(s)$ in (8.13) to obtain the stabilising controller.

The stabilising controller can also be obtained from the frequency domain. The procedure is explained in detail in [31].

8.3: MATLAB Program H-infinity heading control

This is the program used to compute the H-infinity heading controller for the underwater vehicle model, *Subzero II*. There are few stages in the process of designing the H-infinity heading controller. They are: -

1) The plant dynamics and the corresponding weighting function are assigned.
2) If the plant dynamics has poles or zeros on the imaginary axis ($j\omega$), the bilinear shifting axis transformation is employed.
3) The plant and weighting functions are augmented.
4) The H-infinity controller is computed.
5) If any poles or zeros of the plant was shifted earlier, the inverse bilinear shifting axis transformation is employed to the poles or zeros of the H-infinity controller.

The expression for the steady state error in section 5.6 is also computed and different types of plots (figures 5.7 - 5.14) are displayed. Further explanation for the design of the heading H-infinity controller is discussed in chapter 5. The MATLAB robust control toolbox represents system matrices into a single MATLAB variable using the command '*mksys*'. The full program listing to compute the heading H-infinity controller is listed below.

```
%%%

clear all
close all
```

%%% Assigning the transfer function for the plant dynamics and rudder dynamics

% transfer function for the plant dynamics only, G(s)= (-14.1s – 20.95) /(s^3+10.78s^2+15.78s)

```
%no=[-14.1 -20.95];do=[1 10.78 15.17 0];
```

% transfer function for the rudder dynamics, $r_d(s)$=0.9[(7.69)/(s+7.69)]

```
%n1=[0.9*7.69];d1=[1 7.69];
```

*% transfer function for the (plant dynamics * rudder dynamics), $G_r(s)$*

```
ng=[-97.59 -145];dg=[1 18.47 98.07 116.7 0];
n=ng;
d=dg;
```

%convert the transfer function, $G_r(s)$ into state space representation

```
[ag,bg,cg,dg]=tf2ss(n,d);
```

%shift a pole of the plant $G_r(s)$ from 0 to - 0.1 using the bilinear shifting axis transformation

```
[ag0,bg,cg,dg]=bilin(ag,bg,cg,dg,1,'Sft_jw', [inf,-0.1]);
```

%pack matrices describing a system into a single MATLAB variable

```
ssg=mksys(ag0,bg,cg,dg);
```

%%% Assigning weighting functions

%Transfer function of $W_1(s)$: Performance weighting function, $W_1(s) = \frac{5.623s+1}{10s+1e^{-5}}$

```
w1=tf([[(1/0.31)/10^(5/20) 1]],[[(1/0.31) 10^(-100/20)]]);
```

% Transfer function of $W_2(s)$: Control weighting function, $W_2(s)$=1

```
w2=tf([1],[1]);
```

% Transfer function of $W_3(s)$: Robustness weighting function,

```
M=600;  % for high frequency gain
As=0.00001; %for low frequency gain
wbT=6; %bandwidth for T(s)
```

% to form a 2nd order transfer function for $W_3(s) = \frac{s^2+0.06325s+0.001}{0.001667s^2+0.8165s+100}$

```
w3a=[1/sqrt(M) wbT];
w3n=conv(w3a,w3a);
w3b=[1 wbT*sqrt(As)];
w3d=conv(w3b,w3b);
```

```
w3=tf([w3d],[w3n]); %2nd order of W3(s)
```

%%% Augmentation process

```
% augmentation of the plant, Gr(s) and all the weighting functions (W1(s), W2(s), W3(s))
TSS=augtf(ssg,w1,w2,w3);

% compute  the H-infinity controller (sscp) and Tzw (sscl) using the H-infinity control law.
% The variables are represented by a single MATLAB variable: sscp and sscl
[sscp,sscl]=hinf(TSS);

% the H-infinity controller in state space representation
[acp,bcp,ccp,dcp]=branch(sscp);

% shift the resulting pole of the H-infinity control using inverse bilinear transform
[acp,bcp,ccp,dcp]=bilin(acp,bcp,ccp,dcp,-1,'Sft_jw', [inf,-0.1])


% convert the H-infinity controller from the state space representation to transfer function
[num1,den1]=ss2tf(acp,bcp,ccp,dcp);

% the H-infinity controller in transfer function
k=tf([num1],[den1])

% discretised the continuous H-infinity controller using the Tustin (Bilinear) approximation
%method with time sampling of 0.1s

kd=c2d(k,0.1,'tustin')
```

%%% Compute the singular value of T_{zw} (closed loop transfer function)

```
% to create a logarithm horizontal axis from 10^-3 to 10^3
w = logspace(-3,3);

% the Tzw in the state space representation
[acl,bcl,ccl,dcl]=branch(sscl);

% to obtain the singular value of Tzw
svtt = sigma(acl,bcl,ccl,dcl,1,w)

% singular value of Tzw in dB (vertical axis)
svtt = 20*log10(svtt);
```

%%% Steady state error (refer section 5.6 for a brief explanation on steady state error)

```
% plant Gr(s) in transfer function form
gr=tf([n],[d]);

%E(s)=(s*Rs(s))/(1+G(s)K(s)) where E(s) is the error signal, Rs (s)is the desired input signal,
%G(s) is the plant Gr(s) and K(s) is the H-infinity heading controller

G=gr*k;  % open loop
```

```
G2=(1+G);

%Rs (s) is a step input and is assigned as Rs (s)=tf([1],[1 0]). Therefore s*(1/s) = 1, thus the
equation above becomes as below, 1/(1+GK)
E=1/G2


%%% Plots

% Individual W(s)

% Plot of W1(s): Performance weighting function

figure
subplot(2,2,1)
bodemag(w1,{10e-3,10e3})


% Plot of W2(s): Control weighting function

subplot(2,2,2)
bodemag(w2,{10e-3,10e3})


% Plot of W3(s): Robustness weighting function

subplot(2,2,3)
bodemag(w3,{10e-3,10e3})

% Plot of the open loop G(s)K(s)

figure
bodemag(G,{10e-3,10e3})
title('open loop GK')

% Inequalities

S=1/(1+G); % Sensitivity function S(s)
T=G/(1+G); % Complementary sensitivity function T(s)
R=k/(1+G); % Control sensitivity function R(s)

% S(s) < 1/W1(s)

figure
bodemag(S,'b:')
hold on
bodemag(1/w1,{10e-4,10e2})
hold off
title('S < 1/w1')
xlabel('frequency (rad/s)')
ylabel('magnitude (dB)')
```

```
% R(s) < 1/W2(s)

figure
bodemag(R,'b:')
hold on
bodemag(1/w2,{10e-4,10e2})
hold off
title('R < 1/w2')
xlabel('frequency (rad/s)')
ylabel('magnitude (dB)')
% T(s) < 1/W3(s)

figure
bodemag(T,'b:')
hold on
bodemag(1/w3,{10e-4,10e2})
hold off
title('T < 1/w3')
xlabel('frequency (rad/s)')
ylabel('magnitude (dB)')

% Stability margins

% Stability margins for the open loop (G(s)K(s))
figure
wg=logspace(-2,1,1000);
[mag,phase,wg]=bode(G);
margin(mag,phase,wg)

% Stability margins for plant Gr(s)
figure
[mag2,phase2,wg]=bode(gr);
margin(mag2,phase2,wg)

% the H-infinity heading controller

figure
bodemag(k,{10e-4,10e2})


% the singular value of the closed loop transfer function Tzw

figure
semilogx(w,svtt(1,:))
title('Cost Function Tzw ')
xlabel('Rad/Sec')
ylabel('db')
```

8.4 Frequency response for heading error

8.4.1 Data *c01* and *c02*

The data *c01* and *c02* were obtained from the Haslar tank test by Lea [48].

% data for c01	38.4	38.9	39.9
	39	39.1	40.2
39.5	39.1	39.6	39.9
40	38.7	39.6	39.7
39.9	38.7	39.5	39.7
40.1	38.4	39.5	39.3
39.9	37.7	39	40.2
39.5	37.7	39.1	40.1
39.4	37.9	39.4	39.7
39.2	38.2	40.1	39.8
39.9	38.6	40.2	39.1
40.2	40	40	39.4
40.6	41.2	39.9	39.8
40.3	38.7	39.6	39.7
40.2	38.7	39	39.6
39.7	38.3	39.8	38.8
39.5	38.4	40.1	39.1
39.9	38.3	40.2	39.3
40	38.3	39.7	
39.8	38.6	39.6	
40.2	39	39.2	
40	38.6	39.8	
40.3	38.5	40	
39.8	38.7	39.9	
39.2	38.7	40	
41.3	38.6	39.5	
39.6	38.7	39.6	
39.9	39.4	39.8	
39.8	39.2	40	
39.8	39.1	40.3	
39.9	39.1	40.5	
40.1	38.6	39.8	
39.9	38.6	39.4	
39.2	38.9	39.7	
39.4	39.2	40.4	
39.3	39.4	40.4	
39.2	39.3	40.6	
39.6	38.8	39.7	
39.8	38.7	39.8	
39.1	39	39.7	
39	39.7	40.2	
38.8	39.7	40.2	
38.8	39.3	40.1	
38.2	39	39.7	
38.7	39.2	39.8	

% data for c02	57.1
	59.5
70.5	65.1
59.2	62.8
59.1	67.3
64.9	59.6
70.2	66.2
68.9	65.3
65.9	64.4
72.6	66
67.8	62.8
65	67.5
67.7	66
60.4	66
66	71.7
63.2	67.3
61.8	65.7
61.3	69.7
60.7	60.5
63.3	67.2
64.2	65.9
58.2	71.5
58.1	70.6
61.6	69.9
63.4	70.4
58.4	68.1
61.8	69
62.9	68
62.6	69.7
65.7	67
63	66.8
62.4	76.8
58.6	65.2
62	70.4
64.5	70.6
60.9	59.4
62.3	65.3
61.6	62.4
57.6	70.8
57.3	66.1
60.8	70.1
59.6	62.7
61.8	65.9
56.5	63.5
58.8	63.5
61	60.8
63.2	62
59.8	64.9
60.4	
58.2	
63.1	
63.7	
60.2	
65.1	
63.1	

8.4.2 MATLAB Program: Frequency response for heading error

This program is used to obtain the frequency response for the relative heading error of data *c01* and *c02*. The fast Fourier transform method was used to transform these data from time domain to frequency domain. A sampling frequency of 10 Hz was used and the reasons of using 10 Hz as the sampling frequency are given in section 5.3.3. Some considerations were taken into account to the output of the fast Fourier transform. The absolute value was taken and the data were normalised. Only half of the data were taken initially because the data produced after FFT is symmetrical. The data was later multiplied by two to retain the symmetrical property of the original data. However, the first point (1) which is the DC component and the Nyquists frequency component ($N/2$) are unique and doubling these points were not required.

There are two subprograms to calculate the frequency response of data, PlotFFT and a_p_dft. PlotFFT was taken from the MATLAB mathworks website where as a_p_dft was designed by Dr. Antonio de Stefanos. Both subprograms produced similar frequency response result as shown in figures 5.5 and 5.6. Both subprograms were listed in the main program.

```
clear all
close all

%%% Data

% measured heading data obtained in 8.4.1.
hm = [ ];

% desired heading data: 21.67degrees for data c01 and 11.87 degrees for data c02
hd = [ ];

% subtract by 21.67 degrees for data c01 and 11.87 degrees for data c02. Command below is
% an example for data c01
hm(:)=hm(:) – 21.67;

% heading error, he = measured heading - desired heading
he = hm-hd;

% relative heading error, hr= Heading error / desired heading
hr = he/hd;

% sampling frequency of 10 Hz
fs=10;
```

```
%%% FFT (fast Fourier transformation) process

% If want to use subprogram 1: PlotFFT

%%%  from MATLAB technical notes  %%%

% to obtain the frequency response for the heading error. See below for the subprogram 1
%details.
PlotFFT(hr,fs);

%%%  plots  %%%

% frequency in radian per seconds (rad/s)
w=2*pi*f;

figure

% to calculate the maximum value MX in decibels (dB)
MX_db=20*log10(MX);

% plot figure MX (dB) versus frequency (rad/s)
plot(w,MX_db)


% If want to use subprogram 2: a_p_dft

% number of points, N or length of data
%N=1024;

%A=magnitude of the fft output, PHI=phase of the fft output, f= frequency vector
%[A,PHI,f]=a_p_dft(hr,fs,N);

% magnitude of hr in dB
%X=20*log10((A));

% frequency vector in rad/s
%f=2*pi*f;

%plot diagram magnitude (dB) versus frequency (rad/s)
%plot(f,(X))
```

Subprogram 1

This program listing is taken from MATLAB technical notes website. It was derived from Technical Note 1702. For further detail on the structural comments of the program listing, please refer to http://www.mathworks.com/support/tech-notes/v5/1700/1702.shtml. For simplicity, the program listing is attached below. Instead of using the length of the data (length x) and zero padding for sampling from the original program listing, I have given a fixed number of points used for sampling, NFFT = 1024. This is to improve the approximation from

the continuous data to the digitised data. For normalisation, I have divided the magnitude of *hr* (MX) with the number of points NFFT = 1024. While the original program listing divides the magnitude of *x* data by the length of *x* data.

PlotFFT

```
function PlotFFT(x,Fs);

% PLOTFFT: Plot the FFT of a signal. It takes as arguments the signal and the sampling
frequency, and plots the FFT in a figure window.

% PlotFFT(x,Fs): Plots the magnitude of the FFT of the signal x with sampling frequency Fs

% Nyquist frequency
Fn=Fs/2;

%number of points for sampling. Use zero padding if not in power of 2, (2^x)
%NFFT=2.^(ceil(log(length(x))/log(2)));

%give fixed number of points for sampling instead of using the above command
NFFT=1024;

% Take fft, padding with zeros, length(FFTX)==NFFT
FFTX=fft(x,NFFT);
NumUniquePts = ceil((NFFT+1)/2);

% fft is symmetric, throw away second half
FFTX=FFTX(1:NumUniquePts);

% Take magnitude of X
MX=abs(FFTX);

% Multiply by 2 to take into account the fact that we threw out second half of FFTX above
MX=MX*2;

%%% Account for endpoint uniqueness
%DC component of hr
MX(1)=MX(1)/2;

%Nyquist frequency component of hr
MX(length(MX))=MX(length(MX))/2;

% We know NFFT is even. Scale the FFT so that it is not a function of the length of x.
%MX=MX/length(x);

% for normalisation
MX=MX/NFFT;

% frequency vector
f=(0:NumUniquePts-1)*2*Fn/NFFT;

%plot diagram unit versus frequency
plot(f,MX);
```

Subprogram 2

a_p_dft

```
function [A,PHI,f]=a_p_dft(x,fs,N);
% calculate the amplitude (A) and phase (PHI) of the Fourier transform of x.
% if x is real, only the positive frequencies will be given

% if number of argument input is 2 (i.e nothing assigned for N) then N is taken as length(x).
if nargin==2
N=length(x);
end

%fft data x
X=fft(x,N);

% the whole N points data/length
f=[0:N-1]*fs/N;

%Find indices of nonzero elements
i=find(f>fs/2);

%'negative frequencies region'
f(i)=f(i)-fs;
   [f,i]=sort(f); %Sort in ascending order
   X1=X(i);
if sum(imag(x).^2)>0
   disp('WARNING: complex x')
else % for real f, give only the positive frequencies
   f0=find(f==0);
   X2=X1(f0-1:-1:1); % get the negative half of the spectrum
   X1=X1(f0:length(X1)); % get all the positive frequencies
   X1(2:length(X2)+1)=X1(2:length(X2)+1)+conj(X2); % add the negative and positive
frequency components

   f=f(f0:length(f)); % get all the positive frequencies
end;
   A=abs(X1)/N; %normalise by N if value of N is given
   PHI=angle(X1);
 % keyboard
```

8.5 *Subzero II* simulation layout

The layout of the block diagram used in *Subzero II* nonlinear underwater vehicle simulation program [47] is presented in this section. Layout for both H-infinity and PID controls are shown in the following diagrams.

8.5.1: The set up for the H-infinity control

8.5.2: The layout for the PID control

In general, there are three subsystems in both layouts. The three subsystems are: speed control, heading control and depth control. Each of the layout can be illustrated as the block diagram shown below in figure A.

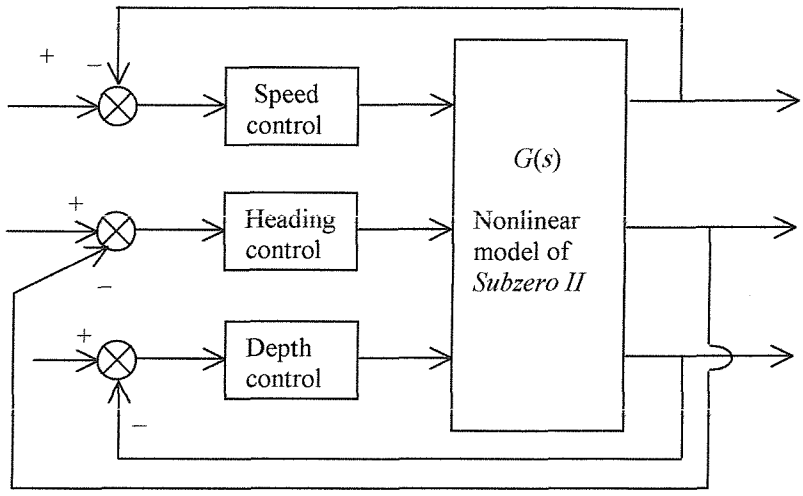


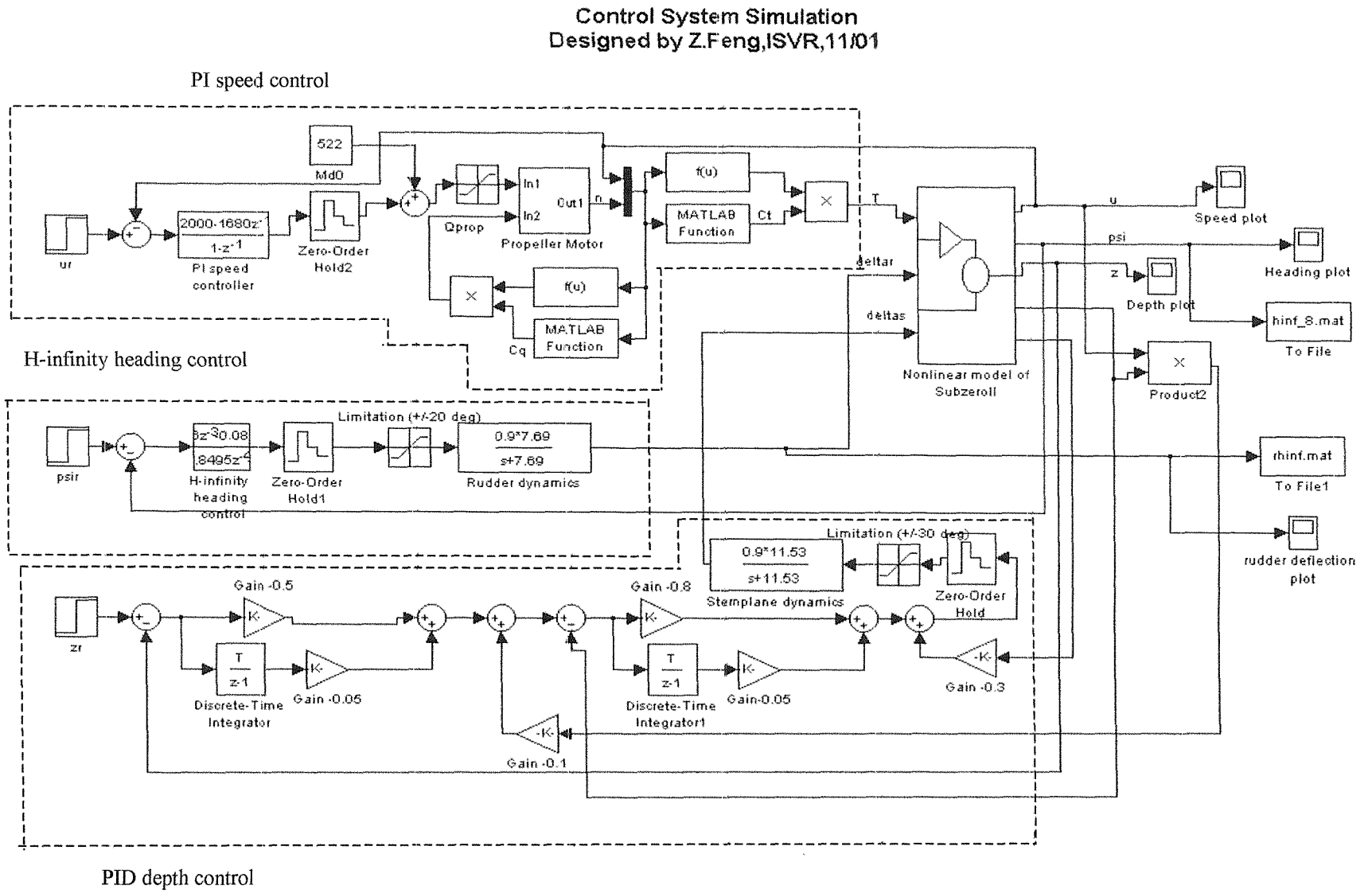
Figure A: General block diagram for *Subzero II* simulation program

Both layouts presented in 8.5.1 and 8.5.2 are almost similar except that the heading control in 8.5.1 is an H-infinity controller while the heading control in 8.5.2 is a PID controller. The control systems for the speed and depth controls for both layouts are PI and PID controls, respectively. Further details on the PI speed and PID depth subsystems can be found in Lea [1]. The input signals are desired speed (*ur*), desired heading (*psir*) and desired depth. (*zr*). These values are given by the *designer*. The three control inputs to the full nonlinear model of *Subzero II* are: DC motor command, (*T*), desired rudder deflection *(deltar)*, and desired sternplane deflection, *(deltas)*. The

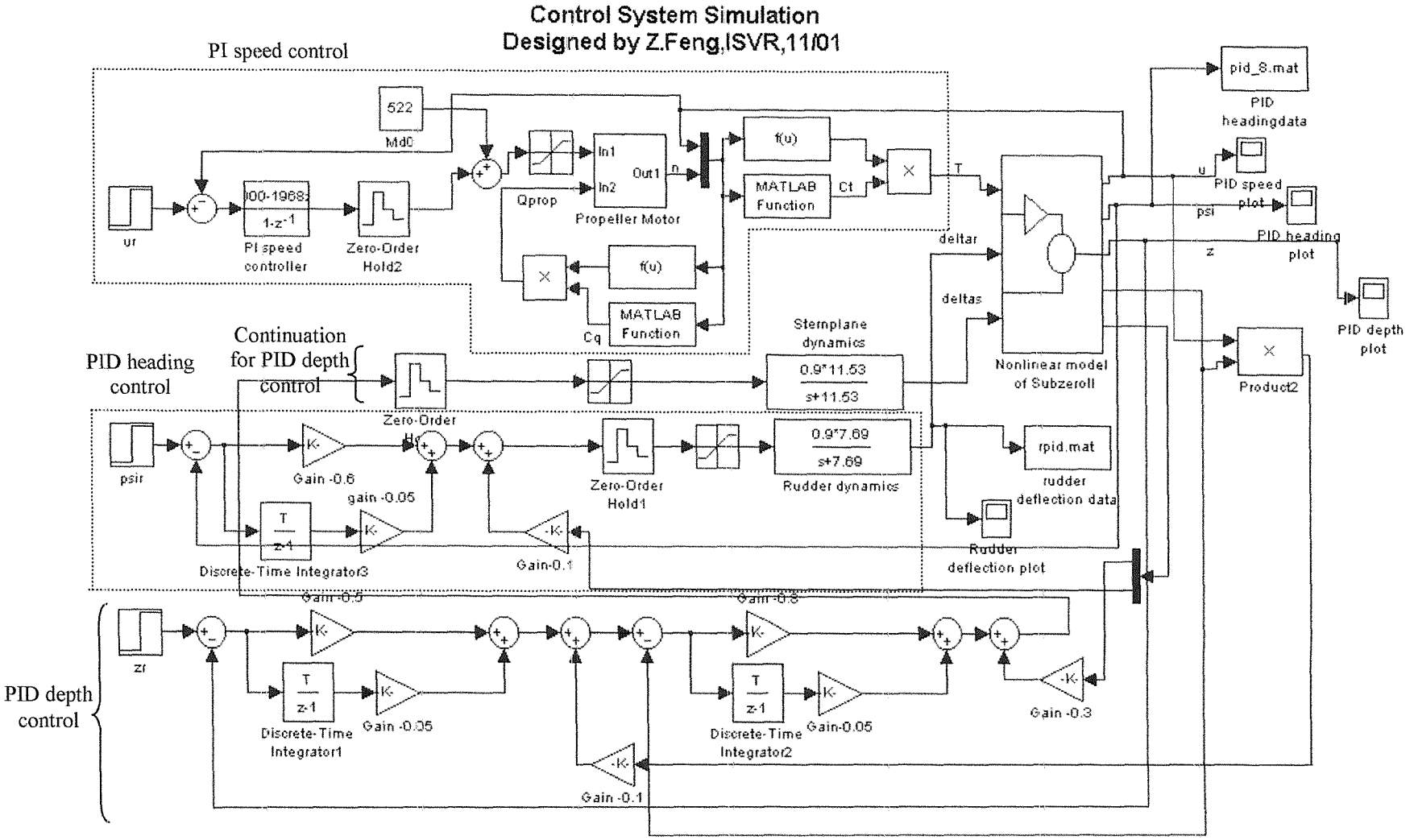
output signals from the plant (nonlinear model of *Subzero II*) are $u\ v\ w\ p\ q\ r\ x\ y\ z\ \phi\ \psi\ \theta$ (but only three states, u (speed) ψ (heading) z (depth) are shown in figure A).

The H-infinity heading controller (5.16) was included into the heading control in layout 8.5.1. Other parameters such as the desired heading, sampling time, were also assigned. The simulation program was run and the heading response can be viewed from the heading plot. The data for the measurement heading can be obtained from file *hinf_8.mat*, while the data for the rudder deflection can be determined from file *rhinf.mat*. Other data and plot can also be obtained by selecting the appropriate states.

8.5.1 The set up for the H-infinity control



8.5.2 The layout for the PID control



8.6 Subprogram for desired H-infinity rudder deflection

This is the subprogram from *AutoROV* used to compute the desired rudder deflection for the H-infinity controller obtained in (5.16). The expression for the desired H-infinity rudder deflection is as below:

$$\begin{aligned}\delta r_{dn} = {} & 1.753*\delta r_{dn-1} + 0.5098*\delta r_{dn-2} - 2.37*\delta r_{dn-3} + 0.8266*\delta r_{dn-4} + 0.6855*\delta r_{dn-5} \\ & - 0.4863*\delta r_{dn-6} + 0.08074*\delta r_{dn-7} \\ & - 0.02985*\ \psi_e - 0.001576*\psi_{en-1} + 0.06648*\psi_{en-2} - 0.001811*\ \psi_{en-3} - 0.04793*\psi_{en-4} \\ & + 0.00625*\psi_{en-5} + 0.01119*\psi_{en-6} - 0.002969*\psi_{en-7}\end{aligned} \tag{8.14}$$

where ψ_e = heading error as the input and δr_d = desired rudder deflection as the output

.

Due to the position of the propeller shaft, the deflection of the rudder is limited to ± 20 degrees. Expression (5.16) is the digitised (z-domain) form of the analogue transfer function in expression (5.13). The z^{-1} term represents a delay of one time step in a discrete-time domain. The expression (8.15) above is the difference equation model [30]. The n^{th} is the latest sample and the $(n\text{-}1)^{th}$ is the previous sample. Therefore, in expression (8.15) above, the current desired rudder deflection (δr_{dn}) is dependent on the values of desired rudder deflection and the heading error of previous sampling intervals, δr_{dn-i} and $\psi_{en-i,}$ respectively, where i = 1, ..., 7. In addition, the current desired rudder deflection also depends on the current heading error.

Only the heading or course control flight was modified to calculate the rudder deflection for the H-infinity control. The speed and the depth control flights remained unchanged. Detail comments for the speed and the depth control flights can be obtained in [12]. The complete subprogram for the flight control is listed below.

```
/* Roy Lea  15/9/97 */
/* File: rov_hinf.c Version: 2.1  */
/* HINF control stuff! */

#include <math.h>
#include <stdlib.h>
```

```
#include <stdio.h>

#include <sub.h>
#include <sim.h>
#include <opt.h>
#include <rov_ext.h>

#if(CONTROL_TYPE==HINF)

double speed_integrator=0.0;
double heading_integrator=0.0;
double depth_integrator=0.0;
double pitch_integrator=0.0;
double z_error_kminus1=0.0;

void hinf_flight_control (SIM_CONTROL *ctrl, STATE *s) {
        void hinf_surge_speed_control (SIM_CONTROL *ctrl, STATE *s);
        void hinf_course_control (SIM_CONTROL *ctrl, STATE *s);
        void hinf_depth_control (SIM_CONTROL *ctrl, STATE *s);

        hinf_surge_speed_control (ctrl, s);
        hinf_course_control (ctrl,s);
        hinf_depth_control (ctrl,s);
        }


void hinf_surge_speed_control (SIM_CONTROL *ctrl, STATE *s) {

        double u_error, n_dot_commanded, gain;
        double K=2000;
        double Ki=3200;
        int i,j;

        u_error = ctrl->speed - sen.speed;

        ctrl->RPS = K*u_error+Ki*speed_integrator;
        if (ctrl->RPS>2100.0) ctrl->RPS=2100.0;
        else if (ctrl->RPS<-2100.0) ctrl->RPS=-2100.0;
        else speed_integrator+=0.1*u_error;
        s->RPS_kminus1 = ctrl->RPS;


                /* Modified subprogram for H-infinity heading response*/


void hinf_course_control (SIM_CONTROL *ctrl, STATE *s) {

/*declare variables*/
double head_diff(double a, double b);
double psi_error;

/*heading error as the input to the desired rudder deflection formula above (8.15) */
psi_error=head_diff(ctrl->course,sen.heading);
```

```
/* Calculate rudder deflection from equation 8.15 */

ctrl->deltar = 1.753*(s->deltar_minus1) + 0.5098*(s->deltar_minus2) - 2.37*
(s->deltar_minus3) + 0.8266*(s->deltar_minus4) + 0.6855*(s->deltar_minus5) - 0.4863*
(s->deltar_minus6) + 0.08074*(s->deltar_minus7)
- 0.02985* (psi_error) - 0.001576*(s->psi_error_minus1) + 0.06648*(s->psi_error_minus2)
- 0.001811* (s->psi_error_minus3) - 0.04793*(s->psi_error_minus4) + 0.00625*
(s->psi_error_minus5) + 0.01119*(s->psi_error_minus6) - 0.002969*(s->psi_error_minus7)


/*Limitation for the rudder due to the shaft linkage to the propeller*/

if (ctrl->deltar>20*TORAD) ctrl->deltar=20*TORAD;
else if (ctrl->deltar<-20*TORAD) ctrl->deltar=-20*TORAD;


/*the value of deltar at the present time will be deltar _minus1 at the next sampling interval*/

s->deltar_minus8 = s->deltar_minus7;
s->deltar_minus7 = s->deltar_minus6;
s->deltar_minus6 = s->deltar_minus5;
s->deltar_minus5 = s->deltar_minus4;
s->deltar_minus4 = s->deltar_minus3;
s->deltar_minus3 = s->deltar_minus2;
s->deltar_minus2 = s->deltar_minus1;
s->deltar_minus1 = ctrl->deltar;

/*the value of psi_error at the present time will be psi_error _minus1 at the next sampling*/
/*interval*/

s->psi_error_minus8 = s->psi_error_minus7;
s->psi_error_minus7 = s->psi_error_minus6;
s->psi_error_minus6 = s->psi_error_minus5;
s->psi_error_minus5 = s->psi_error_minus4;
s->psi_error_minus4 = s->psi_error_minus3;
s->psi_error_minus3 = s->psi_error_minus2;
s->psi_error_minus2 = s->psi_error_minus1;
s->psi_error_minus1 = psi_error;

}
```

/*End of modified subprogram for H-infinity heading response*/

```
void hinf_depth_control (SIM_CONTROL *ctrl, STATE *s) {

 double K=-0.8;
        double Ki=-0.05;
        double Kd=-0.3;
        double z_error, theta_demanded, pitch_error;

        z_error = ctrl->depth - sen.depth;
        theta_demanded = -0.5*z_error
                         -0.05*depth_integrator
                         -0.1*(sen.pitch*sen.speed);
```

```
if (theta_demanded>40.0*TORAD) theta_demanded=40.0*TORAD;
else if (theta_demanded<-40.0*TORAD) theta_demanded=-40.0*TORAD;
else if (fabs(z_error<1.0)) depth_integrator+=0.1*z_error;

pitch_error=theta_demanded-sen.pitch;
ctrl->deltas = K*pitch_error
              +Ki*pitch_integrator
              +Kd*(pitch_error-s->z_dash_kminus1)/0.1;

if (ctrl->deltas>30.0*TORAD) ctrl->deltas=30.0*TORAD;
else if (ctrl->deltas<-30.0*TORAD) ctrl->deltas=-30.0*TORAD;
else if (fabs(pitch_error<10.0*TORAD)) pitch_integrator+=0.1*pitch_error;

z_error_kminus1 = z_error;
s->deltas_kminus1 = ctrl->deltas;
s->z_dash_kminus1 = pitch_error;
}
```

8.7 Uncertainty

An uncertainty is a parameter described as the difference between the actual plant dynamics and the nominal model, G_m. Several uncertainty models can be designed such as the additive and the multiplicative [30],[31]. These uncertainty models are shown in figures B and C below.

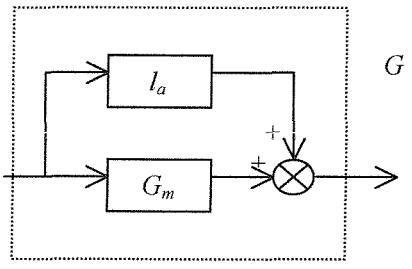


Figure B: Additive uncertainty

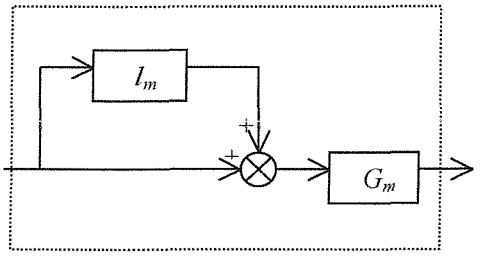


Figure C: Multiplicative uncertainty

l_a and l_m are the uncertainties for the additive and multiplicative models, respectively. The actual plant is $G = G_m + l_a$ for the additive uncertainty plant and $G = G_m\ (1 + l_m)$ for the multiplicative uncertainty plant. The latter model is the most widely applied. This is because it accounts for errors and neglected high frequency dynamics or neglected zeros in the right half s-plane (non-minimum phase zeros). The type of uncertainty model used depends on the stability margins required [31].

Examples of uncertainties are listed below.

a) actuator uncertainties at the plant input
b) unmodelled high frequency dynamics of the plant
c) measurement and sensor uncertainty at the plant output and
d) disturbances at the plant input.

These examples of uncertainty can be represented as additive, multiplicative or parametric uncertainty models. A parametric uncertainty model describes the actual representation of the perturbation. For example, a time delay is normally described as an exponential in the time domain. The form of uncertainty representation depends on one's understanding and knowledge of the uncertainty as well as the system.