Docking of Underwater Vehicle
Model, Autopilot, and Guidance

Anand Sundaresan
Docking of Underwater Vehicle
Model, Autopilot, and Guidance

MASTER OF SCIENCE THESIS

For the degree of Master of Science in Systems and Control at Delft University of Technology

Anand Sundaresan

March 30, 2016

Faculty of Mechanical, Maritime and Materials Engineering (3mE) · Delft University of Technology
The work in this thesis was supported by Allseas Engineering BV. Their cooperation is hereby gratefully acknowledged.
The undersigned hereby certify that they have read and recommend to the Faculty of Mechanical, Maritime and Materials Engineering (3mE) for acceptance a thesis entitled

**Docking of Underwater Vehicle**

by

**Anand Sundaresan**

in partial fulfillment of the requirements for the degree of

**Master of Science Systems and Control**

Dated: March 30, 2016

**Supervisor(s):**

prof.dr.ir. R. Babuška

dr.ir. E. Trottemant

**Reader(s):**

prof.dr.ir. J. Hellendoorn

dr.ir. S. Baldi
Allseas engineering BV is an offshore company that uses underwater vehicles for conducting subsea operations. The two primary classification of such underwater vehicles are: Remote operated vehicle (ROV) and Autonomous underwater vehicle (AUV). As the names suggest, the former requires a human operator to control and the latter is a fully autonomous vehicle. As of now, the company (and most of the offshore industries) customarily use ROVs. In the recent past, Allseas was inclined to use AUVs as a measure of reducing the operational costs and conducted tests runs with an industrial grade AUV. But the test runs were unsuccessful and the plan of using such an AUV was dropped. It was concluded that the major problem was in launch and recovery operations of the AUV, which were conducted from the deck of a ship. To be specific, the disturbances from the ship’s thrusters, ocean currents, and waves, were proved to be impossible to compensate for by the AUV during the launch and recovery operations. Therefore, this thesis aims to investigate on underwater docking capabilities of AUVs, which not only eliminates the launch and recovery issues but improves AUV’s overall operational capabilities. In this thesis, a docking problem is formulated and the solution to the problem covers the following aspects: modeling of AUV, motion control of AUV, and guidance strategies.

For the motion control, an appropriate model of the vehicle is necessary. Various models used in the literature were studied which include: 6 degree of freedom (DOF) non-linear model, 3-DOF horizontal plane model, and 3-DOF vertical plane model. As an example, a 6-DOF non-linear model of ARIES AUV from literature was decoupled into the respective 3-DOF models. These models are used for controller design and docking strategies.

A linear parameter varying (LPV) frame work based, gain scheduled feedback and feed forward controllers were developed for the control of vehicle heading and depth. The controller design involves the following steps: Firstly, a third order Quasi- LPV control plant for heading and depth were derived from horizontal and vertical plane models. Then, linear controllers (gains) were designed for fixed values of scheduling variables. Based on the dimension of the scheduling variables, a stability preserving interpolation of these gains were performed.
to obtain the final controller. A PI controller was designed for controlling the longitudinal velocity of the vehicle. The performance of the controllers were checked on 6 DOF and 3 DOF models of the ARIES AUV.

Finally, two docking strategies: Three point and N-point, were developed using lookahead based path following and investigated their performance for two scenarios: stationary dock and non-stationary dock. The docking strategies addressed the docking problem in two aspects: geometric path generation and path following. Finally, the controller performance in closed loop, as in, the guidance, tracking controllers, and vehicle dynamics was compared for the two developed strategies. It was shown that the N-point docking strategy yielded better convergence to the paths than the three point docking.

**Keywords:** Autonomous underwater vehicle, Remote operated vehicle, Linear parameter varying, and Quasi-linear parameter varying.
Table of Contents

Acknowledgements v

1 Introduction 1
   1-1 Background .................................................. 1
   1-2 Autonomous underwater vehicle .............................. 1
   1-3 Motivation .................................................... 2
   1-4 Problem statement ........................................... 4
   1-5 Scope of the project ......................................... 5
   1-6 Thesis outline ............................................... 6

2 Underwater Docking Systems 9
   2-1 Overview of underwater docking systems ...................... 10
   2-2 Significance of underwater docking for Allseas ............. 13
   2-3 Revisiting the problem statement ............................. 14
      2-3-1 Parameters of interest ................................. 14
      2-3-2 Docking problem: ....................................... 15

3 Modeling of Underwater Vehicle 17
   3-1 Kinematics ................................................... 17
   3-2 Dynamics ..................................................... 19
      3-2-1 Rigid body dynamics .................................... 19
      3-2-2 External forces and moments ............................ 20
      3-2-3 Complete 6 DOF model .................................. 23
   3-3 Horizontal plane model (3 DOF) ............................. 23
   3-4 Model of ARIES AUV .......................................... 25
      3-4-1 Horizontal plane nonlinear model ....................... 28
      3-4-2 Vertical plane nonlinear model ......................... 29
# Table of Contents

4 Autopilot Design .................................................. 31
  4-1 Preliminaries .................................................. 31
    4-1-1 Distinction between LPV and Quasi-LPV systems .......... 32
    4-1-2 Non-linear systems to LPV/Quasi-LPV systems .......... 33
    4-1-3 Linear controllers at frozen points ................. 34
    4-1-4 Stability preserving interpolation ............... 35
  4-2 Non-linear systems to Quasi-LPV systems ............... 39
    4-2-1 Heading angle control plants (Quasi-LPV) ........ 40
    4-2-2 Dive control plants (Quasi-LPV) ................. 41
  4-3 Heading angle control ...................................... 41
    4-3-1 Linear controllers .................................. 41
    4-3-2 Interpolation ....................................... 44
  4-4 Depth control ................................................ 44
    4-4-1 Linear controllers .................................. 45
    4-4-2 Interpolation ....................................... 45
    4-4-3 Controller switching ................................ 47
  4-5 Illustrative example of speed control ................. 47
  4-6 Simulation results .......................................... 48

5 Guidance Algorithms for Docking Maneuver .................. 51
  5-1 Ingredients for guidance algorithms .................... 51
    5-1-1 Way points tracking ................................ 51
    5-1-2 Path following ..................................... 52
  5-2 Docking maneuvers ........................................... 54
    5-2-1 Three point docking maneuver ...................... 54
    5-2-2 N-point docking maneuver ........................... 57
    5-2-3 Docking maneuver for non-stationary dock ........ 57
  5-3 Simulation results ........................................... 58
    5-3-1 Path following ..................................... 58
    5-3-2 Case one: Stationary docking station ............. 60
    5-3-3 Case two: Non-stationary docking station .......... 61

6 Conclusions and Recommendations .......................... 71
  6-1 Conclusions .................................................. 71
  6-2 Recommendations ............................................ 72

Bibliography ......................................................... 75
Acknowledgements

This master thesis was a long journey in which I learnt several technical as well as life lessons. I wholeheartedly thank both of my supervisors, prof.dr.ir. Robert Babuška and dr.ir. Eric Trottemant for their patience and kindness. It wouldn’t have been possible to finish this thesis without their support and guidance. I learnt a lot from both of them and I would like to give one quote from each of them which I might never forget.

From my professor: *Always write down your ideas and findings. It helps organize your thoughts and stay sharp.*

From Eric: *Always approach a problem by taking one step at a time.*

Apart from that, I would like to thank Eric who helped me a lot in asking the right questions to myself and guided me while investigating those questions.

I would like to thank all my friends in Tu Delft, who have been like a family away from home. Finally I would like to thank my parents and brothers for their unimaginable support during the toughest times.

Delft, University of Technology

March 30, 2016

Anand Sundaresan
Chapter 1

Introduction

1-1 Background

Allseas Engineering BV, an offshore pipe lay company, is currently using Remote Operated Vehicle (ROV) for assisting its pipelay operations. ROV is a type of underwater robot that is controlled by a human operator. An ROV is tethered to a support ship by a cable that relays control signals and power, down to the vehicle and returns images and other sensor data to the support ship. It is particularly valuable when the location of the undersea destination is uncertain, or when the ocean conditions endanger a manned mission. But the reliability and robustness of these vehicles are overshadowed by the limitations in the operating region, which is due to the length of the tethering cable. For Allseas, pipelay operations requires the vehicle to be operated at long distances from the pipe laying ship. Moreover, a separate vessel such as survey boats is used to operate these robots. The requirement of support vessel makes the entire ROV operation expensive. Also, ROV needs continuous monitoring from the operator, which in turn increases the operating cost and the requirements for crew support. Therefore, Allseas is inclined to use an Autonomous Underwater Vehicle (AUV), which overcomes the disadvantages of using an ROV. Unlike ROV, AUVs wholly rely on its own autonomy to conduct underwater mission. Also, they are not connected to a support ship through tether, which originally limits the operating range of an underwater robot.

However, the company’s hope on using an AUV considerably plummeted after testing Alistar 3000, an AUV developed by ECA Robotics for assisting pipelay operations in rough offshore environments. There were several problems associated with the unsuccessful test runs, of which launching and recovering the vehicle from pipelay vessel was claimed to be the most challenging one.

1-2 Autonomous underwater vehicle

Autonomous underwater vehicles (AUVs) are intelligent robots which can perform an underwater mission autonomously by maneuvering itself without any human intervention; while
making various decisions using its on-board intelligence and plethora of sensors. AUVs have been of great interest to researchers as it has always been a challenging task to conduct an underwater mission autonomously. The reason is that, the behavior of the vehicle must be carefully programmed to tackle any unanticipated event and for the time varying environment. Therefore, the intelligence of an AUV defines its capability to conduct a mission.

Unmanned Underwater Vehicles (UUVs) include remotely operated vehicles (ROVs) and autonomous underwater vehicles (AUVs) [1]. Based on the maneuvering capabilities, AUVs can further be classified as cruising vehicles and hovering vehicles. The former is used for surveys, search, object delivery, and object location, which run continuously during their mission; whereas, the latter is used for detailed inspection and physical work on and around fixed objects, which hold stations in the water column to perform their tasks [2]. Furthermore, based on the propulsion type, the AUVs can be classified as sea gliders and electrically powered. The former is an emerging technique for propulsion that uses a variable buoyancy principle [3].

AUVs range in size from man portable lightweight AUVs to large diameter vehicles of over 10 meters length. The larger the vehicle is, the greater will be its endurance, sensor carrying capacity, and operating depth. Smaller vehicles benefit significantly from lower logistics, for example, simpler launch and recovery system. Smaller AUVs are at times used as communication gateways between other AUVs and the dock.

In the recent years there have been many designs of AUV shapes such as torpedo, rectangular, biomimetic\(^1\), etc. However, for AUVs with greater operational depths, the torpedo based design is used as it is the best compromise between size, usable volume, hydrodynamic efficiency, and ease of handling. In 1-1, Teledyne’s gavia AUV is shown as an example of the torpedo structure.

\(^1\)Designs and models that are inspired by biological systems.

Figure 1-1: AUV with a torpedo structure.

1-3 Motivation

In 2010, test missions were conducted by Allseas and ECA Robotics from allseas’s Audacia pipe lay vessel. In these test missions, the recovery operation was conducted in the following sequence: The AUV reaches to the surface of water once it has completed all it’s mission,
then the control is taken over by a human operator. The operator maneuveres the vehicle into a floating structure which is tethered to a pipelay vessel. As the vehicle is driven into the structure, a latching mechanism is initiated to lock the vehicle with the structure. Finally the structure which houses the AUV is lifted using on board cranes and winches.

Significant problems were experienced by the crew while recovering the AUV from the water. In the report summary [4], it was shown that the AUV could not even reach near the ship due to the disturbance caused by the thrusters of the ship, ocean currents, and waves. At some point the AUV got sucked towards the pipelay vessel due to the thrusters, which incurred some serious damages on the AUV. It is important to note the extremities in this problem. That is, the AUV is either completely washed away from the pipe lay vessel or it gets pulled towards the pipe lay vessel. The AUV was finally recovered using a support vessel.

As the disturbances from the thrusters of the ship was predominant among other disturbances, it is logical to study its magnitude and range. In Figure 1-2, the disturbance induced by the thrusters of a ship is shown. For a pipe lay vessel that is equipped with dynamic positioning systems, each thruster can rotate $360^\circ$, which essentially means that there will be multiple thruster wash regions\(^2\) around a pipe lay vessel and those regions will not be same at all time instances. Note that, Figure 1-2 depicts only one such instance. Even at 300 meters behind the thrusters, the water currents are approximately 3 knots (1.5 m/s). Whereas, the maximum speed of a AUV is usually around 3-5 knots. Apart from the disturbances from the ships, ocean currents and waves will also contribute to the total disturbance on the AUV. Consequently, compensating such high magnitudes of disturbances during a recovery operation becomes impossible.

\(^2\)The region of disturbance around a thruster.
Some arguments that were considered:

**Why not switch off the thrusters of the ship during a recovery operation?**

Switching off the thrusters of a pipe lay vessel during recovery of AUV is not a feasible option as the vessel uses these thrusters to dynamically position itself while holding a pipe line that extends several kilo meters away from the vessel. Moreover, the position of the pipelaying vessel during pipe lay, determines the path of the pipe in sea bed and switching off the thrusters would risk the vessel to lay pipes in a path that is not prescribed.

**Why not estimate the thruster wash profile based on the orientation of each thruster to come up with a suitable path of entry for the AUV?**

This option is not feasible for several reasons:

1. One cannot guarantee that such a suitable path will exist where the thruster wash can be compensated.
2. Even if such a path exists, it would require the crew to perform the recovery operation at different locations on the deck of the vessel which poses logistical problems.
3. The effect of seas state and weather conditions can cause additional disturbances.

After taking the above discussions into consideration it can be understood that conducting the recovery operations on the pipelay vessel poses severe practical problems. Motivated by [5], [6] and the references there in, this thesis aims to suggest underwater docking as a solution to completely circumvent some of the aforementioned problems if not all. Therefore, autonomous underwater docking of AUV is the starting point of this thesis work.

1-4 **Problem statement**

Autonomous docking is a well researched area in aerospace research community. However, the research and existing technologies of autonomous docking for underwater scenario is still in infant stage [7]. Not much of a research work can be found in the literature addressing the underwater docking problem [7] [8].

The focus of this thesis is to investigate on control and guidance strategies to achieve successful docking of a given Autonomous underwater vehicle (AUV) into a docking station. To this end, given the initial position and orientation of AUV and dock in the three dimensional world, how to achieve successful docking? (See Figure 1-3)

In most of the research work on underwater docking, the docking structure/ station is considered to be fixed at a location and the problem is dealt as a single maneuver to the dock. Most of these research work involves scientific research institutes that are interested in building fixed underwater labs or stations to constantly collect oceanographic data. However, in this project such an assumption would be highly unrealistic as the pipe lay vessel will not be stationary. In simple words, the application of AUV and the necessary docking station is
required to be operated at rough conditions. Addressing the fact that the dock need not be stationary, we use Allseas’s pipe lay process to add more detail to the problem statement. That is, the dock is probable to move a fixed distance at fixed time instances when the pipe lay vessel installs a segment of pipe. Therefore, we consider following two scenarios:

**Dock is stationary**: The position and orientation of the dock is fixed.

**Dock is not stationary**: A step change in dock’s position and orientation.

![Docking Problem Diagram](image)

**Figure 1-3**: Docking problem.

### 1-5 Scope of the project

Any fully autonomous machine will consist of multiple layers of control hierarchy in it, which primarily defines the behavior of the machine in a physical environment; an AUV is no exception to this. That is, there will be multiple levels of control hierarchy each playing a crucial role for a single maneuver. This makes one question the area of focus of this thesis.

In most of the AUV control architectures, one can find three primary levels of categorization in control algorithms: High level, medium level, and low level control [9] [2]. The high
level control is responsible for initiating, supervising, and terminating a given mission. The medium level controller are usually responsible for generating the reference trajectories to be tracked in order to achieve a certain task. Path following, way point tracking, roundabout maneuvers, etc. are some of the examples. The low level controllers make sure that the given reference trajectories are being followed. Relating this project work in this context, we assume that a high level controller exists, which has initiated a docking sequence and focus on the medium and low level controller designs. And hence, in the problem statement 'guidance and control strategies' translates to medium and low level controllers respectively. See Figure 1-4

In this project we consider the AUV to be underactuated mainly because most of the AUVs are designed in such a way as they are primarily used for survey purposes. The model of ARIES AUV developed by naval postgraduate school is used to simulate a docking scenario.

1-6 Thesis outline

This thesis report consists of six chapters, which are organized in the following fashion:

In Chapter 1, the motivation for this thesis project and the problem statement are defined.
In Chapter 2, a comprehensive literature study on underwater docking systems is elucidated in a coherent way. Most importantly, a docking problem is formulated by taking the literature and the requirements of Allseas into consideration.

In Chapter 3, the focus is on general dynamic and kinematic models of an AUV, followed by the decoupling of 6 DOF models into 3 DOF models. Then, the model of ARIES AUV is presented, which is eventually used for the control and guidance strategies.

In Chapter 4, reference tracking controller designs for the parameters that are relevant to the docking problem are given. Important discussions on stability and controller performance for reference tracking are also given.

In Chapter 5, the guidance algorithms are developed that are responsible for generating reference signals for the controllers. The controllers and guidance algorithm works in synergy to perform a docking maneuver. Two docking strategies are developed and analyzed for the docking scenarios defined in problem statement.

In Chapter 6, conclusions of this work are given and possible directions for the future work are suggested.
Chapter 2

Underwater Docking Systems

This chapter serves the purpose of relating the problem statement defined in the previous chapter with the docking problem as a whole. In the following sections, a relevant literature study on underwater docking system is presented. Then, a brief discussion on the significance of such an underwater docking system for Allseas is given. Finally, the problem statement defined in the previous chapter is translated into a control problem.

As such, Autonomous underwater vehicles (AUV) are highly sophisticated vehicles that can perform an underwater mission with maximum autonomy. Yet, these systems are not being readily used due to severe practical limitations such as: tedious launch and recovery operations, limited supply of power, requirement of personnel while conducting a mission, etc. A underwater docking capability eliminates most of these practical limitations if not all. With an underwater dock, the AUV can be deployed for extended times, data can be uploaded and instructions can be downloaded, and the batteries can be recharged; All of which not requiring the vehicle to be recovered from the water.

In the literature, some of the works specifically deal with the docking maneuvers, whereas, some of them take homing maneuvers into consideration. For the readers who are not familiar with these terminologies, the following definitions are given with the intention to clarify the difference in docking and homing maneuvers.

**Docking:** As the name suggests, underwater docking maneuver is the final maneuver of the AUV to dock into an underwater structure that is either stationary or in motion. Such a maneuver is usually initiated 50 -500 meters from the dock.

**Homing:** Homing is the penultimate maneuver to docking. These maneuvers are primarily used to guide the vehicle, so that it is positioned within the operating range of the sensors placed on the dock. Such a maneuver is usually initiated 500 meters -2 Kms from the dock. Once the communication between the dock and vehicle is established, the docking maneuver is initiated.
2-1 Overview of underwater docking systems

In [7], the authors acknowledge that the solutions to docking problem in underwater scenario is still in infant stage while compared to docking problem in aerospace and they discuss about the possibilities in using some of the space docking technologies for underwater docking. Such arguments are also given in [10]. An autonomous docking using Ultra Short Base Line (USBL) \(^1\) is developed in [8] for REMUS (Remote Environmental monitoring unit) AUV by Oceanographic systems laboratory at Wood Hole Oceanographic Institute (WHOI). In this work, all aspects of docking is covered. That is, the dock, charging and communication circuitry, AUV navigation, the vehicle software and the docking algorithm. The dock is conically shaped in order accommodate small magnitudes of cross track error\(^2\) incurred by the approaching vehicle. The dock is shown in Figure 2-1.

The docking maneuver happens in the following sequence: First the vehicle navigates to a position 50 meters from the dock along the track into the dock. Once that criteria has been met, the vehicle attempts to follow the path leading into the dock. When the vehicle determines that it is entering the dock it straightens the fins out, and continues thrusting at constant RPM for 15 seconds. This period forces the vehicle all the way into the guide tube. In this work, no mathematical model of the AUV nor the dock was considered. They extended their work in [11] and [12].

![Docking system for REMUS AUV](image)

**Figure 2-1:** Docking system for REMUS AUV [11].

An electromagnetic homing system was proposed in [13] for a torpedo shaped AUV named SeaGrant Odyssey IIb to dock inside a stationary conical shaped docking station. The docking station was fitted with systems that can emit magnetic fields which the AUV detects

---

\(^1\)USBL is a method of underwater acoustic positioning. A complete USBL system consists of a transceiver, which is mounted on a pole under a ship, and a transponder/responder on the sea floor or in an underwater vehicle.

\(^2\)Distance between the vehicle and a given path
and uses for docking. The proposed system could achieve a precision of up to 20cm to the dock. The docking sequence can be described as follows: The AUV was programmed to travel outbound from the launch point for 60s, execute a 180° turn to point towards the dock station, and travel back toward the ship and dock. The AUV remained in dead reckoning until the magnetic field from the dock could be detected. There is no other communications between the vehicle and the dock. The system was limited to a range of 25 to 30 m. The proposed docking scheme failed when the AUV was aligned more than 30° from the dock.

In [14], a terminal optical guidance based docking technique was demonstrated on two AUVs: a SeaGrant Odyssey IIB and the NRaD Flying Plug. In this approach, the vehicle tracks a source of light from the dock. The terminal guidance term is coined for the final maneuver ranging from 100-10 m from the dock. The control of the vehicle was achieved using a conventional closed loop PID control. They obtained reliable acquisition range of 20-28m according to the rate of turbidity. The optical guidance scheme proved to be effective as the vehicle could dock into the docking station with an accuracy on the order of 1cm at nominal speeds of 1 to 1.5 m/s. One of the main disadvantages of this approach was its susceptibility to the sunlight when docking in shallow waters. Also, the range of the system was considerably altered by the water turbidity.

In [15], a omni directional docking system based on USBL is presented for both homing and docking maneuvers. That is, unlike any of the previous approaches, a pole was used as a docking station, which requires only simple guidance schemes. The authors claim that the presented approach for homing is independent of the initial bearing of the dock to the AUV, includes a method for detecting when the vehicle has missed the dock, and automatically ensures that the AUV is in a position to retry homing with a greater chance of success. A passive latch on the AUV and a pole on the dock accomplishes the task of mechanically docking the vehicle. They developed a layered hierarchical control architecture for the autonomous control of the AUV during the docking procedures using a high level Finite-State Machine (FSM) model to monitor and supervise the whole operation. The system was divided into three different states in a higher level of abstraction:

1. **State 1:** Dock and Vehicle Ready for Mission
2. **State 2:** Vehicle in Mission, Dock Station Empty
3. **State 3:** Vehicle Returns to Dock Station

The docking algorithm for the AUV involved a Line-of-Sight (LOS) technique. An ultra-short baseline system on the vehicle uses this signal to calculate an azimuth and elevation to the dock. The LOS method works by nullifying the bearing to the dock. The heading control law was of the inner-outer loop type, wherein a traditional PID (the inner loop) ensured that the actual vehicle heading \( \psi \) followed the desired heading \( \psi_d \).

This work is particularly interesting as it deals the docking problem in almost every aspect and ascertains that the guidance algorithm for docking is just a tiny part of the docking solution.

Another similar work that covers both docking as well as homing can be seen in [5]. The steps in homing and docking given in the paper is given below:

---

Master of Science Thesis

Anand Sundaresan
1. **Locate and home to the dock.** The vehicle uses its on-board navigation to transit within USBL range of the beacon, about 2 km. The USBL is directional with a $85^\circ$ field of view. The vehicle then homes to the dock using pure pursuit guidance, where the heading control system continuously points the vehicle at the beacon. This guidance strategy does not compensate for ocean current and thus the vehicle can be blown downwind while approaching. Pure pursuit guidance has the advantage of keeping the USBL pointed at the beacon for maximum signal strength.

2. **Compute a position fix.** When the USBL attains good signal strength, the vehicle uses its compass heading and the USBL bearing and range to compute a position fix. This occurs between 1000 and 200 m away from the dock. The position fix is later used in the docking maneuver.

3. **Fly to the start of the final approach path.** The approach path is along the cone centerline, and begins about 300 m out. The approach may require the vehicle to turn away from the dock and temporarily lose USBL contact.

4. **Execute final approach.** The vehicle approaches along the cone centerline using a cross-track controller instead of pure pursuit. This means that the vehicle will track the approach path to the cone, and acquire a drift correction angle if the ocean current has a lateral component. This type of control uses both range and bearing from the USBL, and uses a doppler velocity logger and compass to dead reckon between USBL updates. The vehicle slows to 1.0 m/s about 200 m from the dock. This allows time for the control loop to zero the cross-track error, and also prevents the vehicle from hitting the dock with too much force.

5. **Latch.** The vehicle uses the inductive position sensor and Ethernet contact to determine if it has fully entered. The algorithm then raises the peg and latches the vehicle.

The dock is shown in Figure 2-2.

![Figure 2-2: Stationary Dock used in [5].](image-url)
In [16] and [17], a control solution is presented by integrating both homing and docking as one problem. Conventional artificial potential field method is used for trajectory generation. Sliding mode control is used for robust motion control. Line of sight control is used to make the AUV follow the path. They use the model of a AUV given in [18]. But interestingly, they take 5 DOF model instead of 6 DOF model. The reason is understandable as the equation that represents the roll of the vehicle is not necessary. Therefore, they use three equations for translational velocity but only two equations for angular velocity which are pitch and yaw rates. The motion controller is decoupled into two subsystems of heading and depth (as given in [19]).

In [20], a two step docking approach is presented by integrating the guidance and control strategy for an intervention type AUV developed for the European Union project TRIDENT. The approach to the base, and hence the control design, is divided in two steps: (i) in the first, at higher speed, the vehicle dynamics is assumed to be underactuated, and an appropriate control law is derived to steer the vehicle towards the final docking path, achieving convergence to zero of the appropriate error variables for almost all initial conditions; (ii) in the second stage, at low speed, the vehicle is assumed to be fully actuated, and a robust control law is designed that achieves convergence to zero of the appropriate error variables for all initial conditions, in the presence of parametric model uncertainty.

Finally in [21], a complete control solution is given for the docking problem associated with underwater vehicles. That is, a hierarchical architecture of control is introduced specifically for the problem. Hybrid automata is used for high level control to supervise and execute basic maneuvers. Medium level controllers execute way point tracking, path following, and round about maneuvers. The low level controllers guarantee tracking of the reference signals generated by medium level controllers. In this approach, the dock was considered to be mounted on a Remotely Operated Vehicle (ROV).

2-2 Significance of underwater docking for Allseas

In [7], it is shown that Remotely Operated Vehicles (ROVs) are customarily used for almost all subsea operations, which significantly adds to the operational expenditure for a given subsea project. Thinking in the same line as the authors of [8],

Quote 1:
For autonomous underwater vehicles to be successful, the risks associated with using an untethered vehicle must be compensated by either operational cost savings or the ability to gather data that could not otherwise be obtained.

Quote 2:
As long as ships and crews must be mobilized for each AUV mission, their utility will be limited.

The above quotes are befitting for Allseas engineering BV as the recovery operation depended on the on-board crew support, which does not serve the original purpose of reduced operational expenditure. Moreover, from the reports [4] and [22], it can be noted that the tested AUV was able to achieve it’s primary goals. Yet, the company did not choose to use this technology due to practical limitations such as: tedious launch and recovery operations.
limited supply of power, and most importantly the requirement of support vessels and support crew to conduct a operation.

The above presented literature could be useful for the company to investigate further more on underwater docking, which would reduce the operational cost for subsea operations such as: touch down monitoring, survey, and handling subsea structures. For a company like allseas which is inclined towards designing and developing indigenous technology, developing an underwater docking system based on some of the simple approaches given in the literature would facilitate better quality of subsea operations at lower costs.

2-3  Revisiting the problem statement

Taking the literature study and the problem faced by allseas explained in Chapter 1 into consideration, we can now define more features to the problem.

2-3-1 Parameters of interest

For underactuated AUV motion control, it is a common approach to consider a temporarily separated planar maneuvers rather than spatially coupled maneuvers (motion of aircrafts for instance) [19][23] [24]. Consider a scenario in which the vehicle is required to execute a maneuver to reach a point in 3D space, by temporarily separated planar maneuvers, we mean to say that the vehicle would execute a maneuver in \(xz\)-plane first and then a maneuver in \(xy\)-plane or vice-versa. However, in spatially coupled maneuvers the vehicle would execute both the maneuvers simultaneously.

To keep the problem simple, the separated planar maneuver approach is considered for the docking problem. Therefore, we are interested in controlling the following three variables of the vehicle:

1. Heading angle
2. Vehicle depth
3. Longitudinal velocity

The heading angle and longitudinal velocity combination is used to execute a maneuver in \(xy\)-plane. Similarly, the vehicle depth and longitudinal velocity combination is used for maneuvering in \(xz\)-plane.

---

\(^3\)Monitoring the position at which a given pipeline touches the sea bed during the process of pipe lay.
2-3-2 Docking problem:

Let us define the AUV and dock’s position and orientation in global reference frame as:

\[
P_a := \begin{bmatrix} x_a \\ y_a \\ z_a \end{bmatrix}, \text{position of AUV.}
\]

\[
P_d := \begin{bmatrix} x_d \\ y_d \\ z_d \end{bmatrix}, \text{position of dock.}
\]

\[
\psi_a : \text{Heading angle of AUV.}
\]

\[
\psi_d : \text{Heading angle of dock.}
\]

Assuming that the dock has no pitch angle, we can define the problem of interest as:

\[
P_a(t) = P_d, \quad t > T_{d_1}, \tag{2-1}
\]

\[
u_a(t) = u_d, \quad t > T_{d_1}, \tag{2-2}
\]

\[
\psi_a(t) = \psi_d, \quad t > T_{d_2}, \tag{2-3}
\]

\[
z_a(t) = z_d, \quad t > T_{d_3}. \tag{2-4}
\]

Where \( u_a \) is vehicle’s longitudinal velocity and,

\[
0 < T_{d_3} < T_{d_2} < T_{d_1}. \tag{2-5}
\]

Two cases:

1. **Stationary dock**: \( P_d \) and \( \psi_d \) are constant.

2. **Non-stationary dock**: Step change in dock position and orientation,

\[
P_d = \begin{cases} P_{d_1}, & t = 0, \\ P_{d_2}, & t = T. \end{cases}
\]

\[
\psi_d = \begin{cases} \psi_{d_1}, & t = 0, \\ \psi_{d_2}, & t = T. \end{cases}
\]

The equations (2-1) to (2-4) expresses a desirable maneuver for docking. We are interested in the following sequence of maneuvers from the AUV for successful docking: First execute a maneuver in the \( xz\)-plane so that the dock and the AUV are in same depth. Then, in the \( xy\)-plane approach the dock while aligning the vehicle along the \( x\)-axis of the dock (in body reference frame). The vehicle speed must be the desired speed \( u_d \) as it reaches dock. The time constraint given in (2-5) plays a crucial role in making this maneuver in the required sequence. Moreover, if \( (T_{d_1} - T_{d_2}) \) is large enough, it ensures that the vehicle is aligned along the \( x\)-axis of the dock much before reaching the dock. Similarly, if \( (T_{d_2} - T_{d_3}) \) is large it ensures that the dock and AUV are in same depth before any heading angle is commanded.

Most of the work in the literature concerns with case 1 (stationary dock). Addressing only stationary dock case would be incomplete for Allseas. The pipelay scenario of Allseas is to...
be included in the docking problem. That is, once a given segment of pipe is welded to the existing pipeline, the pipelay vessel moves a constant distance away to install it (See s-lay method for pipe laying for more detail). It can be assumed that the dock is connected to the pipelay vessel and moves with it. After the pipe lay vessel comes to halt for next batch of welding, the dock would come to halt at new position with new orientation. Hence we are interested in addressing a step change in dock position and orientation during the docking maneuver.
Chapter 3

Modeling of Underwater Vehicle

In the beginning stages of developing a model, it is better to understand "why?" and 'For what reason?' the model is being developed. Although, the focus of this thesis is not towards modeling an AUV as such, it deems important to study the approaches incorporated in the literature to obtain the equations of motion of such vehicles. In this case, models are required to analyze the interactions between an underwater vehicle and a dock during final stages of docking. Consequently, to develop control strategies for safe docking operations.

In this chapter, the general steps followed in modeling the equations of motion of an underwater vehicle are shown. Initially, the kinematics involved in the model is dealt. Then, the rigid body dynamics along with external forces and moments, that constitutes the dynamical model are dealt. Finally, a reduced Degrees of Freedom (DOF) model is derived, which is later used for developing autopilots and guidance algorithms.

3-1 Kinematics

In cases where a vehicle operates in a limited geographical area, it is usually sufficient to use a local flat earth approximation to express the location of the vehicle. Let the global reference frame be the NED frame (North, East, and Down) and it is denoted as \( \{n\} \). The frame \( \{n\} \), is assumed to be inertial. Let \( \{b\} \), denote the frame attached to the vehicle, and hence called as body fixed reference frame. The axes of the frame coincides with the principle axes of the vehicle. For convenience, the origin of \( \{b\} \) frame is chosen to be the center of gravity (CG) of the vehicle, as it reduces the complexity of the model. Let the position of the vehicle in inertial reference frame be represented as, \( P_{nb}^n \). Where, \( P_{nb}^n \in \mathbb{R}^3 \) is the vector from the origin of \( \{n\} \) frame to the origin of the \( \{b\} \) frame, decomposed in the \( \{n\} \) frame. Similarly, the orientation of the vehicle is represented as \( \Theta \in S^3 \), which consists of Euler angles (pitch, roll, and yaw).

Now, the linear and angular velocities of the vehicle relative to the inertial frame, decomposed
in inertial frame can be defined as:

\[ v_{nb}^n := \dot{P}_{nb}^n, \]
\[ \omega_{nb}^n := \dot{\Theta}. \] (3-1)

The velocities in (3-1) and (3-2) can be related to the velocities expressed in \( \{b\} \) frame as:

\[ v_{nb}^n = R_{nb}^b v_{nb}^b, \] (3-3)
\[ \omega_{nb}^n = T_{\Theta} \omega_{nb}^b. \] (3-4)

Where, the rotation matrix \( R_{nb}^b \in SO(3) \) (Special Orthogonal group) is the transformation from \( \{n\} \) to \( \{b\} \), or equally, the transformation from \( \{b\} \) to \( \{n\} \). But, the transformation matrix \( T_{\Theta} \) in equation (3-4), is not a member of \( SO(3) \). That is, \( T_{\Theta}^{-1} \neq T_{\Theta}^T \). The derivation of such a transformation can be seen in [18]. Both, \( R_{nb}^b \) and \( T_{\Theta} \) are parametrized by the Euler angles \( (\phi, \theta, \psi) \). The correspondence between the variables used above and the commonly used notation established by the Society of Naval Architects and Marine Engineers (SNAME) [25] are listed in Table 3-1 and shown in Figure 3-1.

Equations (3-3) and (3-4) can be grouped into matrix form as:

\[
\begin{bmatrix}
    v_{nb}^n \\
    w_{nb}^n
\end{bmatrix} =
\begin{bmatrix}
    R_{nb}^b & 0_{3\times3} \\
    0_{3\times3} & T_{\Theta}
\end{bmatrix}
\begin{bmatrix}
    v_{nb}^b \\
    w_{nb}^b
\end{bmatrix}.
\] (3-5)

\( J(\Theta) \) in (3-5) gives the coordinate transformation of velocities from \( \{b\} \) to the \( \{n\} \).

Figure 3-1: Reference frames [26].
### Table 3-1: Nomenclature

<table>
<thead>
<tr>
<th>Description</th>
<th>Variable</th>
<th>SNAME Entries</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vehicle position in cartesian coordinates</td>
<td>$P^n_b$</td>
<td>$[x \ y \ z]^T$</td>
</tr>
<tr>
<td>Vehicle attitude (Roll, Pitch, and Yaw)</td>
<td>$\Theta$</td>
<td>$[\phi \ \theta \ \psi]^T$</td>
</tr>
<tr>
<td>Linear velocity</td>
<td>$v^b_n$</td>
<td>$[u \ v \ w]^T$</td>
</tr>
<tr>
<td>Angular velocity</td>
<td>$\omega^b_n$</td>
<td>$[p \ q \ r]^T$</td>
</tr>
<tr>
<td>External forces</td>
<td>$f^b$</td>
<td>$[X \ Y \ Z]^T$</td>
</tr>
<tr>
<td>External moments</td>
<td>$m^b$</td>
<td>$[K \ M \ N]^T$</td>
</tr>
</tbody>
</table>

#### 3-2 Dynamics

Modeling the dynamics of a rigid body moving in a viscous fluid is of two folds. First, the motion of the rigid body is modeled based on conservation of linear and angular momenta (Newton’s laws), which is a standard way of modeling in mechanics. Second, the influence of infinite-dimensional dynamics of the fluid around the body is modeled as finite dimensional approximations. Majority of concepts dealt in this chapter are from [18].

##### 3-2-1 Rigid body dynamics

The equations of motion of the rigid body can be written in component form with the following assumptions:

1. The origin of the $\{b\}$ frame coincides with CG of the vehicle.
2. The axes of $\{b\}$ frame coincides with the vehicle’s principle axes of inertia, which makes the moment of inertia matrix to be diagonal.

Now, the equations governing the motion of the body in three dimensional world is given as:

$$
\begin{align*}
\dot{u} &- vr + wq = X, \\
\dot{v} &- wp + ur = Y, \\
\dot{w} &- uq + vp = Z, \\
I_x\dot{p} + (I_z - I_y)qr & = K, \\
I_y\dot{q} + (I_x - I_z)rp & = M, \\
I_z\dot{r} + (I_y - I_x)pq & = N.
\end{align*}
$$

Where, $m$ is the mass of the vehicle. The equations (3-7)-(3-11) can be expressed in vectorial form as shown in [18]:

$$
M_{RB}\ddot{\nu} + C_{RB}(\nu)\nu = \tau_{RB},
$$

Anand Sundaresan
where, $\nu = \begin{bmatrix} v_{nb}^b & \omega_{nb}^b \end{bmatrix}^T$ is the generalized velocity vector decomposed in \{b\} frame and $\tau_{RB} = \begin{bmatrix} f^b & m^b \end{bmatrix}^T$ is the generalized vector of external forces and moments. $M_{RB}$ is the inertia matrix and $C_{RB}(\nu)$ is the matrix consisting coriolis and centripetal terms. The inertia matrix is given as:

$$M_{RB} = \begin{bmatrix}
  m & 0 & 0 & 0 & 0 & 0 \\
  0 & m & 0 & 0 & 0 & 0 \\
  0 & 0 & m & 0 & 0 & 0 \\
  0 & 0 & 0 & I_x & 0 & 0 \\
  0 & 0 & 0 & 0 & I_y & 0 \\
  0 & 0 & 0 & 0 & 0 & I_z \\
\end{bmatrix} = \begin{bmatrix}
  mI_{3 \times 3} & 0_{3 \times 3} \\
  0_{3 \times 3} & I_o \\
\end{bmatrix}. \tag{3-13}
$$

Similarly, the coriolis and centripetal terms can be written as:

$$C_{RB} = \begin{bmatrix}
  0 & 0 & 0 & 0 & mw & -mv \\
  0 & 0 & 0 & mw & 0 & mu \\
  0 & 0 & 0 & mv & -mu & 0 \\
  0 & mw & -mv & 0 & I_z r & -I_y q \\
  -mw & 0 & mu & -I_z r & 0 & I_x \\
  mw & -mu & 0 & I_y q & -I_x p & 0 \\
\end{bmatrix} = \begin{bmatrix}
  0_{3 \times 3} & mS(v_{nb}^b) \\
  mS(v_{nb}^b) & S(I_o \omega_{nb}^b) \\
\end{bmatrix}. \tag{3-14}
$$

where, $S(v_{nb}^b)$ is a skew symmetric matrix and $I_o$ is inertia matrix. Note that, the matrices $M_{RB}$ and $C_{RB}(\nu)$ are in the simplest possible forms due to the assumptions made previously.

For an extensive explanation on the derivations of the equations of motion, the reader is prescribed to consult [18].

### 3-2-2 External forces and moments

The ambiguity in modeling the motion of an underwater vehicle rises from the right hand side of (3-12). This is due to the fact that, there has not been a standard procedure to model these forces and moment as it is vehicle dependent [27]. In fact, these are the major sources for model uncertainty and neglected higher order dynamics. Before the reader proceeds any further, it is important to mention here that, the objective is to model the interactions of surrounding fluids affecting the motion of vehicle. Unlike the previous section, where the equations of motion were derived using Newtonian mechanics, here a Lagrangian based approach is used to model the hydrodynamic effects (or rather the forces) on the vehicle. However, the focus of this thesis is not in modeling a vehicle as such, and so, only an overview of the modeling approach is dealt. The generalized forces and moments which consists of infinite dimensional dynamics are modeled as linear approximations:

$$\tau_{RB} = \tau_S + \tau_H + \tau, \tag{3-15}$$

where, $\tau_S$ is hydrostatic forces and moments, $\tau_H$ is hydrodynamic forces and moments, and $\tau$ is control and propulsion forces.
Hydrodynamic Forces

The generalized hydrodynamic forces arise from the reaction of the surrounding fluid to the vehicle motion. The hydrodynamic forces can be broadly categorized under two groups, one involving the motion of the vehicle in ideal or inviscid fluid and the other, due to fluid viscosity effects.

\[ \tau_H = \tau_I + \tau_R, \]  

(3-16)

where, \( \tau_I \) is forces due to ideal fluids and \( \tau_R \) is forces due to real viscous fluids.

The forces due to added mass are the main contribution to the hydrodynamic forces due to the motion in ideal fluids. It deems important at this juncture to explain the concept of added mass for better understanding.

**Added mass:** Added (virtual) mass can be understood as the pressure induced forces and moments due to the motion of the body, proportional to its acceleration. The forces and moments due to this phenomenon are grouped as:

\[ \begin{bmatrix} X_A & Y_A & Z_A & K_A & M_A & N_A \end{bmatrix}^T. \]  

(3-17)

Where, the subscript "A" denotes added mass. The expression for these forces are derived using Kirchhoff’s approach (Quasi Lagrangian) and fluid kinetic energy. If the linear and angular velocities are grouped into a single vector as:

\[ \nu = \begin{bmatrix} \nu_b \\ \omega_{nb} \end{bmatrix} = \begin{bmatrix} \nu_1 \\ \nu_2 \end{bmatrix}, \]  

(3-18)

then, the kinetic energy possessed by the fluid can be written as:

\[ T_A = \frac{1}{2} \nu^T M_A \nu. \]  

(3-19)

Where, \( M_A \) is the inertia matrix corresponding to the added mass effects. The expression for (3-17) are obtained by Kirchhoff’s equations which are written as:

\[ \frac{d}{dt} \left( \frac{\partial T_A}{\partial \nu_1} \right) + S(\nu_2) \frac{\partial T_A}{\partial \nu_1} = \tau_1, \]  

(3-20)

\[ \frac{d}{dt} \left( \frac{\partial T_A}{\partial \nu_2} \right) + S(\nu_2) \frac{\partial T_A}{\partial \nu_2} + S(\nu_1) \frac{\partial T_A}{\partial \nu_1} = \tau_2. \]  

(3-21)

Derivation of these expressions are out of the scope of his report. However, the derivations finally gives us inertia and Coriolis matrices based on added mass which are shown as:

\[ M_A = - \begin{bmatrix} X_{\tilde{u}} & 0 & 0 & 0 & 0 & 0 \\ 0 & Y_{\tilde{u}} & 0 & 0 & 0 & 0 \\ 0 & 0 & Z_{\tilde{u}} & 0 & 0 & 0 \\ 0 & 0 & 0 & K_{\tilde{u}} & 0 & 0 \\ 0 & 0 & 0 & 0 & M_{\tilde{u}} & 0 \\ 0 & 0 & 0 & 0 & 0 & N_{\tilde{u}} \end{bmatrix}, \]  

(3-22)

\(^1\text{An inviscid flow is the flow of an ideal fluid that is assumed to have no viscosity.}\)
\[
C_A(v) = \begin{bmatrix}
0 & 0 & 0 & 0 & -Z_ww & Y_vv \\
0 & 0 & 0 & Z_ww & 0 & -X_uu \\
0 & 0 & 0 & 0 & -Y_ww & X_uu \\
0 & -Z_ww & Y_vv & 0 & -N_rr & M_qp \\
Z_ww & 0 & -X_uu & N_rr & 0 & -K_qp \\
-Y_vv & X_ww & 0 & -M_qp & K_pp & 0
\end{bmatrix}.
\] (3-23)

The hydrodynamic coefficients in the \( M_A \) and \( C_A(v) \) are written with respect to the SNAME notation. For example, the hydrodynamic added mass force \( Y_A \) along the \( y - axis \) due to an acceleration \( \dot{u} \) in the \( x \)-direction is written as:

\[
Y_A = -Y_u \dot{u},
\] (3-24)

where,

\[
Y_u := \frac{\partial Y}{\partial \dot{u}}.
\] (3-25)

Finally, the forces and moments due to the ideal fluid can be written as:

\[
\tau_I = -M_A \dot{v} - C_A(v).
\] (3-26)

**Damping:** The motion of a rigid body in a viscous fluid causes frictional forces, usually called as drag on the rigid body. In a viscous fluid, the frictional forces are present such that the system is not conservative with respect to energy. In general, these forces have both linear and non-linear terms associated with it. The generalized drag which contributes to \( \tau_R \) is written as:

\[
D(v) = D + d(v),
\] (3-27)

where, \( D \) is the linear component and \( d(v) \) is the non-linear component of drag.

Therefore, the hydrodynamic forces and moments can be added up as:

\[
\tau_H = -M_A \dot{v} - C_A(v) v - D(v).
\] (3-28)

**Hydrostatic forces**

Besides the added mass and damping forces, the underwater vehicle will be affected by gravity and buoyancy forces. Let \( m \) be the mass of the vehicle, \( \nabla \) be the volume of fluid displaced by the vehicle, \( g \) be the acceleration of gravity, and \( \rho \) be the water density. Then, the weight and buoyancy of the vehicle can be defined as:

\[
W = mg, \quad B = \rho g \nabla.
\] (3-29)

The vehicle is assumed to be neutrally buoyant. That, is \( W = B \). Let the distance between the center of gravity \( CG \) and the center of buoyancy (CB) be defined by the vector,

\[
BG = \begin{bmatrix}
x_g - x_b \\
y_g - y_b \\
z_g - z_b
\end{bmatrix}
\] (3-30)

Also, when the CG and Center of Buoyancy (CB) are assumed to be located vertically on the \( z - axis \) then the hydrostatic forces can be written as:

\[
\tau_s = \begin{bmatrix}
0 & 0 & 0 & -z_b W \cos \theta \sin \phi & -z_b W \sin \theta & 0
\end{bmatrix}^T.
\] (3-31)
Control forces

The number of degrees of actuation is vehicle dependent and it cannot be generalized. However, the general classification of control forces are forces generated by the control surfaces and propulsion forces.

\[ \tau = \tau_c + \tau_p. \]  (3-32)

\( \tau_c \) and \( \tau_p \) are forces and moments due to control surfaces and propellers respectively.

Therefore, the generalized forces and moments, \( \tau_{RB} \) in the right hand side of the (3-12) can now be summed into the following equation:

\[ \tau_{RB} = \tau - M_A \dot{\nu} - C_A(v)\nu - D(\nu)\nu - \tau_S. \]  (3-33)

3-2-3 Complete 6 DOF model

Having explained the kinematics and dynamical equations, now it becomes convenient to summarize the entire model. In order to obtain the complete model the rigid body dynamics and external forces are merged together by substituting (3-33) in (3-12). In simple words, the 6 DOF model of an underwater vehicle can be expressed in two matrix equations as:

\[ \dot{\eta} = J(\eta)\nu, \]  (3-34)

\[ M\dot{\nu} + C(\nu)\nu + D(\nu)\nu + g(\eta) = \tau. \]  (3-35)

Where,

- \( M = M_{RB} + M_A \), generalized inertia matrix,
- \( C(\nu) = C_{RB}(\nu) + C_A(\nu) \), generalized coriolis and centripetal terms,
- \( D(\nu) = D + d(\nu) \), generalized damping,
- \( \eta = \left[ \begin{array}{c} P_{\eta_{\eta}} \Theta \end{array} \right]^T \), position and orientation,
- \( \nu = \left[ \begin{array}{c} v_{\eta_{\nu}} \omega_{\eta_{\nu}} \end{array} \right]^T \), linear and angular velocity,
- \( \tau = \) vector of control input.

Note that, (3-34) is the kinematic equation and (3-35) is the dynamic equation. It can be witnessed that the model obtained is highly non-linear and coupled.

3-3 Horizontal plane model (3 DOF)

The objective of the thesis is to optimize the docking maneuver of an underwater vehicle. The docking problem is dealt in horizontal plane which motivates to deduce a reduced degrees of freedom model from the previously obtained 6 DOF model. To study the motion of the vehicle in the horizontal plane, it is sufficient to consider the translations along surge (x-axis) and sway (y-axis), and the rotation about the heave (z-axis). Therefore, we choose
\( \mathbf{\nu} = \begin{bmatrix} u & v & r \end{bmatrix}^T \) and the \( \mathbf{\eta} = \begin{bmatrix} x & y & \psi \end{bmatrix}^T \). This is implies that the dynamics associated with the motion in heave, roll, and pitch are not considered, that is \( w = p = q = 0 \).

The kinematic equation (3-34) reduces to:

\[
J(\mathbf{\eta}) = R(\psi) = \begin{bmatrix} \cos \psi & -\sin \psi & 0 \\ \sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix},
\]  
(3-36)

The rigid body terms reduces as:

\[
M_{RB} = \begin{bmatrix} m & 0 & 0 \\ 0 & m & 0 \\ 0 & 0 & I_z \end{bmatrix},
\]  
(3-37)

\[
C_{RB}(\mathbf{\nu}) = \begin{bmatrix} 0 & 0 & -mv \\ 0 & 0 & mu \\ mv & -mu & I_z \end{bmatrix}.
\]  
(3-38)

Notice that surge is decoupled from sway and yaw motions due to symmetry considerations. Similarly, the added mass terms reduces as:

\[
M_A = -\begin{bmatrix} X_u & 0 & 0 \\ 0 & Y_{\dot{u}} & -Y_{\dot{\psi}} \\ 0 & N_{\dot{v}} & N_u \end{bmatrix},
\]  
(3-39)

\[
C_A(\mathbf{\nu}) = \begin{bmatrix} 0 & 0 & Y_{\dot{r}}r + Y_{\dot{\psi}}v \\ 0 & 0 & -X_{\dot{u}}u \\ -Y_{\dot{r}}r - Y_{\dot{\psi}}v & X_{\dot{u}}u & 0 \end{bmatrix}.
\]  
(3-40)

For 3 DOF model, the surge motions in system inertia matrix becomes decoupled from sway and yaw. This implies that a linear damping is a good assumption for low speed applications. The linear damping matrix can be given as:

\[
D = \begin{bmatrix} -X_u & 0 & 0 \\ 0 & -Y_{\dot{\psi}} & -Y_{\dot{r}} \\ 0 & -N_{\dot{v}} & N_{\dot{r}} \end{bmatrix}.
\]  
(3-41)

And the rest of the terms follows such deduction. The final model in 3 DOF can be written as:

\[
\dot{\mathbf{\eta}} = R(\psi)\mathbf{\nu},
\]  
(3-42)

\[
M\dot{\mathbf{\nu}} + C(\mathbf{\nu})\mathbf{\nu} + D = \tau.
\]  
(3-43)
Where,

\[ M = M_{RB} + M_A, \text{generalized inertia matrix}, \]
\[ C(\nu) = C_{RB}(\nu) + C_A(\nu), \text{generalized coriolis and centripetal terms}, \]
\[ D = \text{Linear damping}, \]
\[ \eta = [x \ y \ \psi]^T, \text{position and orientation}, \]
\[ \nu = [u \ v \ r]^T, \text{linear and angular velocity}, \]
\[ \tau = [\delta_r \ n]^T, \text{rudder angle and propeller RPM}. \]

3-4 Model of ARIES AUV

As explained earlier, the equations of motion of underwater vehicles are vehicle dependent as there are multiple factors which determine the behavior of a vehicle. That is, some vehicles have control in all degrees of freedom whereas, most of the vehicles are under actuated. Moreover, the type of actuation could vary depending on the application. Apart from actuation, the size and form of a vehicle significantly affect the hydrodynamic forces acting on it. Therefore, there is not a single standard model to study the behavior of underwater vehicles. Between the years 1990 to 2000, 30 AUV were developed worldwide whose summary can be seen in [1], to understand the difference in design philosophy and application.

In this thesis, the model of the AUV developed by the Naval Postgraduate School (NPS) is used to simulate and analyze the docking problem. The schematic diagram of the vehicle is shown in Figure 3-2. Information about the vehicle hardware and software can be found in [28]. Also, the hydrodynamic coefficients of the vehicle is available in [19], which considerably influences the motion of the vehicle.

The vehicle has five control inputs: rudder angle \( \delta_r \), stern plane \( \delta_s \), starboard bow plane \( \delta_{sb} \), port side bow plane \( \delta_{bp} \), and propeller rotation rate \( n \).

![Figure 3-2: NPS ARIES AUV [19].](image)
Input, $\tau = \begin{bmatrix} \delta_r & \delta_s & \delta_b & \delta_{bs} & \delta_{bp} & n \end{bmatrix}^T$, \hfill (3-44)

state, $X = \begin{bmatrix} u & v & w & p & q & r & x & y & z & \phi & \theta & \psi \end{bmatrix}^T$, \hfill (3-45)

where,

- $u$ - surge velocity (m/s),
- $v$ - sway velocity (m/s),
- $w$ - heave velocity (m/s),
- $p$ - roll velocity (rad/s),
- $q$ - pitch velocity (rad/s),
- $r$ - yaw velocity (rad/s),
- $x$ - position in x-direction (m),
- $y$ - position in y-direction (m),
- $z$ - position in z-direction (m),
- $\phi$ - roll angle (rad),
- $\theta$ - pitch angle (rad),
- $\psi$ - yaw angle (rad).

In this case, the dynamical equation given in (3-35) can be written with respect to NPS AUV in component form:

**Surge Equation of Motion**

\[
m \ddot{u} - ur + wq = \frac{\rho}{2} L^4 [X_{pp} \dot{p}^2 + X_{qq}\dot{q}^2 + X_{rr}\dot{r}^2 + X_{pr}\dot{p}\dot{r}] + \frac{\rho}{2} L^3 [X_{ppu} + X_{wqp}\dot{q} + X_{vp}\dot{p} + X_{vp}\dot{p}]
+ \int_{x_{nose}}^{x_{tail}} \left( C_{dy} h(x)(v + x\dot{r})^2 - C_{dz} b(x)(w - x\dot{q})^2 \right) \frac{U_{cf}(x)}{U_{cf}(x)} \, dx
+ (W - B) \sin \theta + \frac{\rho}{2} L^3 X_{s}\omega q \delta_s \epsilon(n)
+ \frac{\rho}{2} L^2 [X_{wq}\dot{q} + X_{q}\dot{q}] + \frac{\rho}{2} L^2 u^2 X_{prop}.
\] \hfill (3-46)

**Sway Equation of Motion**

\[
m \ddot{v} + ur + wq = \frac{\rho}{2} L^4 [Y_{pp}\dot{p} + Y_{pp}\dot{r} + Y_{pp}q + Y_{pp}\dot{q}] + \frac{\rho}{2} L^3 [Y_{wq}\dot{q} + Y_{wq}\dot{q} + Y_{wq}\dot{q} + Y_{wq}\dot{q} + Y_{wq}\dot{q}]
+ \int_{x_{nose}}^{x_{tail}} \left( C_{dy} h(x)(v + x\dot{r})^2 - C_{dz} b(x)(w - x\dot{q})^2 \right) \frac{U_{cf}(x)}{U_{cf}(x)} \, dx
+ \frac{\rho}{2} L^3 [X_{wq}\dot{q} + X_{wq}\dot{q} + X_{wq}\dot{q} + X_{wq}\dot{q} + X_{wq}\dot{q}]
+ \frac{\rho}{2} L^2 u^2 X_{prop}.
\] \hfill (3-47)
Heave Equation of Motion

\[ m[\ddot{w} - uq + vp] = \frac{\rho}{2} L^4 \left[ Z_q \dot{q} + Z_{pp} \dot{p}^2 + Z_{pr} \dot{r} + Z_{rr} \dot{r}^2 \right] + \frac{\rho}{2} L^3 \left[ Z_w \dot{w} + Z_q uq + Z_{vp} \dot{p} + Z_{vr} \dot{r} \right] + \frac{\rho}{2} L^2 \left[ Z_{ww} \dot{w} + Z_{vv} \dot{v}^2 + u^2 (Z_{\delta s} \delta s + Z_{\delta b} \delta b + Z_{\delta b / 2} \delta b_p) \right] + \frac{\rho}{2} \int_{x_{tail}}^{x_{nose}} \left[ C_{dy} h(x) (v + x r)^2 - C_{dx} b(x) (w - x q)^2 \right] \frac{(w - x q)}{U_c(x)} \, dx + (W - B) \cos \theta \cos \phi + \frac{\rho}{2} L^3 Z_{wn} uq \epsilon(n) + \frac{\rho}{2} L^2 \left[ X_{w \delta s n} u v \delta s n + X_{\delta s \delta b} u^2 \delta s^2 \right] \epsilon(n). \]

(3-48)

Roll Equation of Motion

\[ I_{x \dot{p}} + (I_z - I_y) \dot{q} = \frac{\rho}{2} L^5 \left[ K_{p \dot{p}} + K_r \dot{p} + K_{pq} \dot{q} + K_{pr} \dot{r} \right] + \frac{\rho}{2} L^4 \left[ K_{\dot{q}} \dot{q} + K_{uq} + K_{vp} \dot{p} + K_{vr} \dot{r} + K_{vpq} \dot{q} + K_{vpp} \dot{p} + K_{vpr} \dot{r} \right] + \frac{\rho}{2} \int_{x_{tail}}^{x_{nose}} \left[ C_{dy} h(x) (v + x r)^2 - C_{dx} b(x) (w - x q)^2 \right] \frac{(w + x q)}{U_c(x)} \, dx + (y_G W - y_B B) \cos \theta \cos \phi - (z_G W - z_B B) \cos \theta \sin \phi + \frac{\rho}{2} L^3 K_{mn} u q \epsilon(n) + \frac{\rho}{2} L^5 u^2 K_{prop}. \]

(3-49)

Pitch Equation of Motion

\[ I_{y \dot{q}} + (I_x - I_z) \dot{r} = \frac{\rho}{2} L^5 \left[ M_{\dot{q}} \dot{q} + M_{pp} \dot{p}^2 + M_{pr} \dot{r} + M_{rr} \dot{r}^2 \right] + \frac{\rho}{2} L^4 \left[ M_{w} \dot{w} + M_{uq} uq + M_{vp} \dot{p} + M_{vr} \dot{r} \right] + \frac{\rho}{2} \int_{x_{tail}}^{x_{nose}} \left[ C_{dy} h(x) (v + x r)^2 - C_{dx} b(x) (w - x q)^2 \right] \frac{(w + x q)}{U_c(x)} \, dx \]

(3-50)

Yaw Equation of Motion

\[ I_{z \dot{r}} + (I_y - I_x) \dot{p} = \frac{\rho}{2} L^5 \left[ N_{\dot{p}} \dot{p} + N_r \dot{r} + N_{pp} \dot{p} + N_{pr} \dot{r} \right] + \frac{\rho}{2} L^4 \left[ N_{\dot{w}} \dot{w} + N_{uq} uq + N_{vpp} \dot{p} + N_{vpr} \dot{r} \right] + \frac{\rho}{2} \int_{x_{tail}}^{x_{nose}} \left[ C_{dy} h(x) (v + x r)^2 + C_{dx} b(x) (w - x q)^2 \right] \frac{(v + x r)}{U_c(x)} \, dx + \frac{\rho}{2} \left( x_G W - x_B B \right) \cos \phi \sin \phi + \frac{\rho}{2} L^3 u^2 N_{prop}. \]

(3-51)
Where,

\[ U_c(x) = (v + x_r)^2 + (w - x_q)^2, \]
\[ X_{prop} = C_{d0}(|\eta| - 1), \]
\[ \eta = 0.012 n/u, \]
\[ C_{d0} = 0.00385, \]
\[ \epsilon(n) = -1 + \frac{\text{sign}(n)}{\sqrt{|C_t + 1|}} - \frac{\text{sign}(u)}{\sqrt{|C_{tt} + 1|}}. \]

The kinematic equation as given in (3-34), can be written in component form as:

\[ \dot{\phi} = p + q \sin \phi \tan \theta + r \cos \phi \tan \theta, \] (3-52)
\[ \dot{\theta} = q \cos \phi - r \sin \phi, \] (3-53)
\[ \dot{\psi} = (q \sin \phi + r \cos \phi) / \cos \theta, \] (3-54)
\[ \dot{x} = u \cos \psi \cos \theta + v [\cos \psi \sin \theta \sin \phi - \sin \psi \cos \phi] + w [\cos \psi \sin \theta \sin \phi + \sin \psi \sin \phi], \] (3-55)
\[ \dot{y} = u \sin \psi \cos \theta + v [\sin \psi \sin \theta \sin \phi + \cos \psi \cos \phi] + w [\sin \psi \sin \theta \sin \phi - \cos \psi \sin \phi], \] (3-56)
\[ \dot{z} = -u \sin \theta + v \cos \theta \sin \phi + w \cos \theta \sin \phi. \] (3-57)

Although the model of NPS AUV provides the best possible information on the dynamics of the vehicle, the model is highly non-linear and coupled. In order to work with simpler models, especially for control design, it has been a practice to decouple the model into three different non-interacting subsystems: speed, steering, and diving models. Note that, the roll mode is assumed to be passive. Such a decoupling is possible by separating the dynamics into horizontal plane and vertical plane.

### 3-4-1 Horizontal plane nonlinear model

A simpler model is obtained by assuming that \( w, p, q, \phi, \theta, \delta_s, \delta_b, \), and \( \delta_{bp} \) to be zero. That is, only the horizontal plane is considered. From Eq. (3-46)-Eq. (3-51), only SurgeEq. (3-46), SwayEq. (3-47), and Yaw Eq. (3-51) are considered. The reduced equations are given as:

**Surge Equation**

\[
(m - a_3 X_u) \dot{u} = \left[ a_3 (X_{ur} + m) \right] vr + (a_4 X_{vr}) \dot{r}^2 + (a_4 X_{rv}) \dot{v}^2, \\
+ (a_3 X_{r\delta_s}) \dot{u} \delta_r + (a_2 X_{v\delta_s}) \dot{u} \dot{v} \delta_r + (a_2 X_{\delta_s \delta_r}) \dot{u}^2 \delta_r^2 + a_2 u \dot{X}_{prop}. \]

**Sway Equation**

\[
(m - a_3 Y_v) \dot{v} - (a_4 Y_r) \dot{r} = \left[ a_3 Y_r - m \right] ur + (r_2 Y_u) uv + (a_2 Y_{\delta_r}) \dot{u}^2 \delta_r. 
\]

Anand Sundaresan Master of Science Thesis
3-4 Model of ARIES AUV

Yaw Equation

\[
(-a_4 N_y) \dot{\psi} - (I_z - a_5 N_r) \dot{r} = (a_4 N_r) ur + (a_3 N_v) uv + (a_3 N_{\delta s}) u^2 \delta_r. \tag{3-60}
\]

The Kinematic equations reduces to:

\[
\begin{aligned}
\dot{x} &= u \cos \psi - v \sin \psi, \\
\dot{y} &= u \sin \psi - v \cos \psi, \\
\dot{\psi} &= r.
\end{aligned}
\tag{3-61, 3-62, 3-63}
\]

The nonlinear equations can be written in the following form:

\[
\begin{aligned}
M \dot{x}_1(t) &= f(x_1(t)) + g(x_1(t), u_1(t)), \\
\dot{x}_2(t) &= h(x_1(t), x_2(t)).
\end{aligned}
\tag{3-64, 3-65}
\]

Where the generalized inertia matrix \(M\) is given as,

\[
M = \begin{bmatrix}
m - a_3 X_u & 0 & 0 \\
0 & m - a_3 Y_v & -a_4 Y_r \\
0 & -a_4 X_\psi & a_5 N_r - I_z
\end{bmatrix}.
\tag{3-66}
\]

The states and inputs pertaining to the model are given as:

\[
x_1 = \begin{bmatrix} u & v & r \end{bmatrix}^T, x_2 = \begin{bmatrix} x & y & \psi \end{bmatrix}^T, u_1 = \begin{bmatrix} \delta_r & n \end{bmatrix}^T.
\]

3-4-2 Vertical plane nonlinear model

The vertical plane model is obtained by assuming that \(v, p, q, r, \phi, \psi, \delta_s, \delta_{bs}, \delta_{bp}\) to be zero. The surge(3-46), heave(3-48), and pitch(3-50) equations are reduced as:

Surge Equation

\[
(m - a_3 X_u) \ddot{u} = (a_3 X_{\dot{u}} - m) \dot{w} + (a_2 X_{\dot{u} \dot{v}}) \dot{w} + (a_3 X_{\dot{u} \dot{q}}) \dot{q},
\]

\[
+ [a_2 X_{\dot{q}} \delta_r + a_3 X_{\dot{q} \delta sn} \epsilon(n)] \dot{u} \delta_s + [a_2 X_{\dot{w}} \delta_s + a_2 X_{\dot{w} \delta sn} \epsilon(n)] \dot{w} \delta_s,
\]

\[
+ [a_2 X_{\dot{v}} \delta_r + a_2 X_{\dot{v} \delta sn} \epsilon(n)] u^2 \delta_s^2 + a_2 u^2 X_{\text{prop}}. \tag{3-67}
\]

Heave Equation

\[
(m - a_3 Z_w) \ddot{w} - (a_4 Z_q) \dot{q} = (a_3 Z_q) \dot{u} + (a_2 Z_w) \dot{w},
\]

\[
+ (a_2 Z_{\delta r}) \dot{u} \delta_s + (a_3 Z_{\delta sn}) \dot{u} \epsilon(n) + (a_2 Z_{\dot{w} w}) \dot{w} \epsilon(n) + (a_2 Z_{\delta sn} a_2 u^2 \delta_s \epsilon(n)). \tag{3-68}
\]
Pitch Equation

\[
(-a_4 M_w) \dot{w} + (I_y - a_5 M_q) \dot{q} = (a_4 M_u q) u + (a_3 M_w w) - z G W \sin \theta,
\]

\[
+ (a_3 M_s) u^2 \delta_s + (a_4 M_q n) u q (n) + (a_3 M_w n) u w (n) + (a_3 M_sn) u^2 \delta_s (n).
\]

(3-69)

The Kinematic equations reduces to:

\[
\dot{x} = u \cos \theta,
\]

(3-70)

\[
\dot{z} = -u \sin \theta,
\]

(3-71)

\[
\dot{\theta} = q.
\]

(3-72)

The nonlinear equations can be written in the following form:

\[
M \dot{x}_1(t) = f(x_1(t)) + g(x_1(t), u_1(t)),
\]

(3-73)

\[
\dot{x}_2(t) = h(x_1(t), x_2(t)).
\]

(3-74)

Where the generalized inertia matrix \(M\) is given as,

\[
M = \begin{bmatrix}
m - a_3 X_u & 0 & 0 \\
0 & m - a_3 Z_w & -a_4 Z_q \\
0 & -a_4 M_w & I_y - a_5 M_q
\end{bmatrix}.
\]

(3-75)

The states and inputs pertaining to the model can be given as:

\[
x_1 = \begin{bmatrix} u & w & q \end{bmatrix}^T, x_2 = \begin{bmatrix} x & z & \theta \end{bmatrix}^T, u_i = \begin{bmatrix} \delta_s & n \end{bmatrix}^T.
\]

Note that the surge equation is different for the horizontal and vertical plane models. These models are relatively simpler and less coupled, and so, used for control design.
The three important parameters of interest in the docking problem presented in Chapter 2, are heading angle, depth, and the surge velocity (longitudinal velocity) of the AUV. The design of reference tracking controller for heading angle, depth, and longitudinal velocity are discussed in this chapter.

4-1 Preliminaries

The autopilot design approach in this thesis is based on gain scheduling. A survey paper [29], gives an overview of design approaches and steps incorporated in the literature of gain scheduling. The four main steps in any gain scheduled controller design are as follows:

1. The first step is to deduce a linear parameter varying (LPV) or Quasi-linear parameter varying (Quasi-LPV) model from the original non-linear model. There are few common approaches to obtain such a model. To name a few: Jacobian linearization, Quasi-lpv descriptions, and linear fractional descriptions are some of the most widely used approaches [29],

2. Design linear controllers at frozen or fixed points of the scheduling variable. That is, controllers are designed for a family of LTI plants that are indexed by the scheduling parameter. As a result we obtain a family of linear controllers, that are indexed by the fixed scheduling parameter. Typically, the controller at these frozen points are designed to meet the desired performance specification.

3. The linear controllers are then patched together in real time by the scheduling signal, which results in a continuous controller. Controller switching or interpolation of controllers are some of the common avenues.

4. Finally, the performance of the controller on the non-linear system is analyzed.
4-1-1 Distinction between LPV and Quasi-LPV systems

Linear parameter varying systems and Quasi-linear parameter varying systems form the basis for gain scheduling [29] [30] [31] [32]. LPV framework is the middle ground between linear and non-linear systems [33]. The main purpose of bringing the non-linear system into the LPV form, is to exploit and use the well known techniques of linear systems. Recall that the first step in the gain scheduling is to deduce the nonlinear systems into LPV or Quasi-LPV systems, and so, it is important to clearly understand the differences in such systems.

**LPV Systems** A linear parameter varying system (LPV) is basically a non-stationary linear system whose dynamics depend upon an exogenous parameter whose values are unknown a priori [33][34]. The exogenous parameter could either be measured or estimated. Typical a priori assumptions on such parameters are bounds on the magnitude of the exogenous parameter ($\rho$) and the rate of variation of $\rho$. A subtle difference between the LPV systems and Linear Time Varying systems (LTV) is that the dynamics of LTV systems depend on the time parameter which is a known priori (as in periodic systems). The LPV system description is expressed in Figure 4-1, where $u$ and $y$ represent control input and output signals respectively. A possible realization of the plant is:

\[
\dot{x}(t) = A(\rho)x(t) + B(\rho)u(t), \tag{4-1}
\]
\[
y(t) = C(\rho)x(t), \tag{4-2}
\]

where, $x$ is the state of the system and $u$ is the control input.

![Figure 4-1: LPV systems [34]](null)

**Quasi-LPV Systems** Quasi-LPV systems are a special case of LPV systems, in which, the exogenous parameter is actually endogenous to the system dynamics [34]. That is, the parameter that schedules the parameterized set of linear systems is part of the system dynamics. For instance, the scheduling parameter in Eq. (4-1) and Eq. (4-2) could be one of the elements of the state vector $x$. When $\rho$ is permitted to depend on state $x$, the dependency of the system matrices $A$ and $B$ on the state introduces non-linear feedback, which is not present in LTI or LTV systems [32]. Therefore, the use of the term linear for describing such nonlinear systems is misleading, and hence, the term quasi is used.

In general, when the scheduling parameter is strictly exogenous, the reformulation of the non-linear systems leads to LPV systems. However, if the scheduling parameter is permitted
to depend on the state, input or output of the system, the resulting reformulation leads to Quasi-LPV systems.

4-1-2 Non-linear systems to LPV/Quasi-LPV systems

In the literature of gain scheduling, there have been numerous approaches to reformulate a given non-linear system into the LPV/Quasi-LPV form. In [32], an overview of most of these approaches are dealt along with their short comings. Some of the approaches are Jacobian linearization, state transformation, velocity based transformation, and output dependent quasi-LPV transformation. Addressing each of these approaches is out of context. However, the two main approaches that is widely incorporated in many applications are shown:

**Jacobian Linearization** Let a nonlinear system be defined as:

\[
\begin{align*}
\dot{x}(t) &= f(x(t), u(t)), \\
y(t) &= h(x(t)),
\end{align*}
\]

where, \(x\) is the state of the system and \(u\) is the control input. A plant equilibrium point \((x_e, u_e)\) is a point such that \(f(x_e, u_e) = 0\). If the system Eq. (4-3) has a family of equilibrium points that are parameterized by the scheduling variable, \(\rho\), then \(f(x_e(\rho), u_e(\rho)) = 0\) is the equilibrium family. Definition of the family of equilibrium points and the selection of scheduling signals are problem dependent. Corresponding to a specified equilibrium family there is a plant linearization family that can be written in the linear parameter-varying form:

\[
\begin{align*}
\dot{x}_\delta(t) &= A(\rho)x_\delta(t) + B(\rho)u_\delta(t), \\
y(t) &= C(\rho)x(t),
\end{align*}
\]

where the system matrices are from:

\[
\begin{align*}
A(\rho) &= \frac{\partial}{\partial x} f(x, u)|_{x=x_e(\rho), u=u_e(\rho)}, \\
B(\rho) &= \frac{\partial}{\partial u} f(x, u)|_{x=x_e(\rho), u=u_e(\rho)},
\end{align*}
\]

and the deviation variables are defined in the obvious fashion:

\[
x_\delta(t) = x(t) - x_e(\rho), \quad u_\delta(t) = u(t) - u_e(\rho).
\]

Note that the scheduling parameter is in turn a function of time, \(\rho(t)\).

**State transformation** This approach is based on the possibility of rewriting the plant in a form where nonlinear terms can be hidden with newly defined, time-varying parameters that are then included in the scheduling variable. In order to explain this approach better, an example from [29] is elucidated:
Example Consider a nonlinear plant,

\[ \dot{x}_1 = \sin x_1 + x_2, \quad \dot{x}_2 = x_1 x_2 + u; \]  

(4-10)

One possible Quasi-LPV form of the plant is:

\[ \dot{x} = A(x)x + Bu = \begin{bmatrix} \sin x_1 / x_1 & 1 \\ x_2 & 0 \end{bmatrix} x + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u. \]  

(4-11)

Note that, in the above Quasi-LPV description, gain scheduling is possible only if both the states are available for state feedback. Now, if only one state is available for feedback, then the plant can be re-written as:

\[ \dot{x} = A(x)x + Bu = \begin{bmatrix} \sin x_1 / x_1 & 1 \\ 0 & x_1 \end{bmatrix} x + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u. \]  

(4-12)

Therefore, deriving a Quasi-LPV model is problem dependent.

4-1-3 Linear controllers at frozen points

Let the linear plant models be parameterized by a scheduling variable, \( \rho \). Then for fixed values of \( \rho_i, i = 1, 2, \ldots q \) a family of LTI systems are obtained as:

\[ \dot{x}(t) = A(\rho_i)x(t) + B(\rho_i)u, \]  

(4-13)

\[ y(t) = C(\rho_i)x(t). \]  

(4-14)

Let the control law:

\[ u(t) = -K_i x(t) + G_i r(t), \]  

(4-15)

yield a closed loop system,

\[ \dot{x}(t) = [A(\rho_i) - B(\rho_i)K_i]x(t) + B(\rho_i)G_i r(t). \]  

(4-16)

It is assumed that the states are measurable and available for feedback. Also, \( (A(\rho_i), B(\rho_i)) \) is stabilizable and \( (A(\rho_i), C(\rho_i)) \) is detectable. The feedback gains are obtained by solving the quadratic cost function containing the states and input of the system:

\[ J = \int_0^\infty x^T(t)Q_i x(t) + u^T(t)R_i u(t)dt, \]  

(4-17)

where, \( Q_i \geq 0 \) and \( R_i > 0 \) are tuned to render \( [A(\rho_i) - B(\rho_i)K_i] \) Hurwitz \( \forall \rho_i, i = 1, 2, \ldots q \). Now, the steady state response of the controlled system Eq. (4-16) is,

\[ y = -C(\rho_i)[A(\rho_i) - B(\rho_i)K_i]^{-1}B(\rho_i)G_i r. \]  

(4-18)

In order to make \( y = r \), \( G_i \) is chosen such that \(-C(\rho_i)[A(\rho_i) - B(\rho_i)K_i]^{-1}B(\rho_i)G_i = I \). Such a \( G_i \) is obtained by solving the following regulator equation:

\[ \begin{pmatrix} A(\rho_i) & B(\rho_i) \\ C(\rho_i) & 0 \end{pmatrix} \begin{pmatrix} \Pi \\ \Gamma \end{pmatrix} + \begin{pmatrix} 0 \\ -I \end{pmatrix} = 0, \]  

(4-19)
where, $\Pi = -(A(\rho_i) - B(\rho_i)K_i)^{-1}B(\rho_i)G_i$ and $\Gamma = G_i - K_i\Pi$. Alternatively, if $x_{ss}$ and $u_{ss}$ are the states and input at steady state, then,

$$x_{ss} = \Pi r,$$

$$u_{ss} = \Gamma r.$$  \hspace{1cm} (4-20)

Of course, the resulting feed forward gains by solving the regulator equation achieves tracking for constant reference signals. For slowly varying the reference signals, it a usual practice to add integrator to the system. This is done by augmenting an extra state:

$$\begin{bmatrix} \dot{x}_I \\ \dot{x} \end{bmatrix} = \begin{bmatrix} 0 & C \\ 0 & A(\rho) \end{bmatrix} \begin{bmatrix} x_I \\ x \end{bmatrix} + \begin{bmatrix} 0 \\ B(\rho) \end{bmatrix} u + \begin{bmatrix} 0 \\ 1 \end{bmatrix} r.$$  \hspace{1cm} (4-22)

### 4-1-4 Stability preserving interpolation

The interpolation techniques are from [35]. The important lemmas and definition from [35] are explained below to justify that the controller synthesis preserves stability. Initially, we consider multiple gains stabilizing a single LTI plant and then we arrive at the argument that the interpolation of all those gains can preserve stability for the plant. Then we extend this argument for multiple LTI plants with the use of a stability covering condition. The type of interpolation in the synthesis of the final continuous controller depends on the dimension of the scheduling variable. It will be shown later in the chapter that we are interested in scalar scheduling variable and two dimensional vector scheduling variable cases only.

**Lemma 3.1** Suppose that $K_1, \ldots, K_q$ stabilize $(A, B)$, that is, each $(A - BK_i)$ is stable. Then there exist symmetric positive-definite matrices $W_1, \ldots, W_q$ such that for any constants $\mu_1, \ldots, \mu_q$, not all zero, the state feedback gain,

$$K = (\mu_1 K_1 W_1 + \cdots + \mu_q K_q W_q)(\mu_1 W_1 + \cdots + \mu_q W_q)^{-1},$$  \hspace{1cm} (4-23)

stabilizes $(A, B)$.

**Remark 3.1** If $K_1$ and $K_2$ both stabilize $(A, B)$, then by Lemma 3.1, there exist symmetric positive-definite matrices $W_1$ and $W_2$ such that for every fixed $\mu, 0 \leq \mu \leq 1$

$$K = (\mu K_1 W_1 + (1 - \mu) K_2 W_2)(\mu W_1 + (1 - \mu) W_2)^{-1},$$

stabilizes $(A, B)$. Additionally if $W_1 = W_2$, then $K = \mu K_1 + (1 - \mu) K_2$ stabilizes $(A, B)$. This provides sufficient condition for linear interpolation to preserve stability.

**Definition** Given $A(\rho)$ and $B(\rho)$ as in Eq. (4-1), $\rho \in \Lambda$, suppose the state feedback gain $K_i$ is such that $A(\rho_i) + B(\rho_i)K_i$ is stable, $i = 1, \ldots, q$. Let $U_i$, containing $\rho_i$, be an open neighborhood such that $A(\rho) + B(\rho)K_i$ is stable for each fixed $\rho \in U_i$. If $\Lambda \subset \bigcup_{i=1}^q U_i$ then the state feedback gains are said to be satisfying the *stability covering* condition.
Case 1: Scalar scheduling variable  The continuous control law resulting from interpolation of linear controllers based on scalar scheduling variable is an adaptation from the theorem given in [35], which uses the lemma 3.1 and the stability covering condition. As given in the stability covering condition, for a given fixed scheduling variable, an open neighborhood region is established in which, a single feedback gain can stabilize all the control plants within the neighborhood region. Now, for any two adjacent points in the scheduling variable set, the neighborhood regions are chosen in such a way that they intersect with each other. If a single point is chosen within the region of intersection of neighborhood regions, using Remark 3.1, one can guarantee that both gains at $\rho_i$ and $\rho_{i+1}$ will stabilize the control plant corresponding to the point. Hence the region of intersection is chosen to be the region of interpolation where stability can be preserved. The pictorial idea of neighborhood regions and the interval of interpolation is shown in Figure 4-2. The interval of interpolation is $[b_i, c_i] \subset U_i \cap U_{i+1}$, $i = 1, \ldots, q - 1$. The resulting continuous control law is given as:

$$K(\rho) = \begin{cases} 
K_i, & \rho \in [\rho_i, b_i), \\
K_i(\rho)W^{-1}, & \rho \in [b_i, c_i], \\
K_{i+1}, & \rho \in (c_i, \rho_{i+1}]. 
\end{cases} \quad (4-24)$$

Where,

$$\tilde{K}(\rho) = \frac{c_i - \rho}{c_i - b_i}K_iW_i + \frac{\rho - b_i}{c_i - b_i}K_{i+1}W_{i+1},$$

$$W(\rho) = \begin{cases} 
W_i, & \rho \in [\rho_i, b_i), \\
W_i + \frac{\rho - b_i}{c_i - b_i}W_{i+1}, & \rho \in [b_i, c_i], \\
W_{i+1}, & \rho \in (c_i, \rho_{i+1}]. 
\end{cases} \quad (4-25)$$

Note that the above interpolation is linear interpolation. The reader is advised to relate the coefficients $\frac{c_i - \rho}{c_i - b_i}$ and $\frac{\rho - b_i}{c_i - b_i}$ with $\mu_i$ and $\mu_{i+1}$ as given in lemma 3.1 and Remark 3.1. Instead of interpolation one can simply switch controllers when the scheduling variable lies in the intersection region. For certain plants, controller switching can cause sudden jumps in the control signal that might not be desirable. Controller interpolation is preferred in those situations.

Figure 4-2: Neighborhood regions and interpolation intervals.
Case 2: Vector scheduling variable  
Extending the scalar scheduling variable case, we can now implement the same concepts for vector scheduling variables. Let the vector scheduling variable be defined as:

\[
\rho = \begin{bmatrix} \rho_1 \\ \rho_2 \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix}.
\]

In the allowable set for the scheduling variable, \( \Lambda \), the linear controllers are designed at the grid points of the space. That is, there are feedback and feed forward gains at each of the grid points and we are interested in obtaining the gains at points other than the grid points through interpolation while preserving stability.

Consider a single block from a 2D grid to understand the interpolation approach (See Figure 4-3). The gains \( K_{11}, K_{21}, K_{12}, K_{22} \) at the vertices of the square will render the corresponding control plant stable. Subsequently, there exists positive definite matrices \( W_{11}, W_{21}, W_{12}, W_{22} \) that can satisfy the Lyapunov equation (For example, \( W_{11}[A-BK_{11}]^T + [A-BK_{11}]W_{11} < 0 \)). To obtain a gain at a point inside the square, we need to interpolate the gains at the four vertices. Such an interpolation leads to Bilinear interpolation, which is an extension to linear interpolation. Bilinear interpolation is a two step linear interpolation, one along the \( x-axis \) and the other along the \( y-axis \), or vice versa. In Figure 4-3, linear interpolation along \( x-axis \) results in gains \( K_1, K_2 \). Then, the interpolation along the \( y-axis \) between \( K_1 \) and \( K_2 \) results in the final \( K \).

Let us define the open neighborhood regions \( U_{11}, U_{21}, U_{12}, U_{22} \) corresponding to the gains. In order to make use of the stability covering condition, the neighborhood regions must intersect with each other. The region of bilinear interpolation is the region of intersection of all the four neighborhood regions, where the control plant corresponding to a fixed scheduling variable can be stabilized by all the four gains. Consequently the interpolation of the gains will still preserve stability (from Lemma 3.1). The region of linear interpolation is the intersection of two consecutive neighborhood regions (as shown in scalar case). The controller resulting from bilinear interpolation can be written as:

\[
\begin{bmatrix} dy_2 \\ dy_1 \end{bmatrix}_{K_1(\rho)} \begin{bmatrix} K_{11}W_{11} & K_{21}W_{21} \\ K_{12}W_{12} & K_{22}W_{22} \end{bmatrix} \begin{bmatrix} dx_2 \\ dx_1 \end{bmatrix}_{W_{1}(\rho)} + \begin{bmatrix} dy_2 \\ dy_1 \end{bmatrix}_{W_{1}(\rho)} \begin{bmatrix} W_{11} & W_{21} \\ W_{12} & W_{22} \end{bmatrix} \begin{bmatrix} dx_2 \\ dx_1 \end{bmatrix},
\]

where

\[
\rho \in U_{11} \cap U_{21} \cap U_{12} \cap U_{22}.
\]

(4-26)
Figure 4-3: Neighborhood regions and interpolation region.

The controllers resulting from linear interpolation can be written as:

\[
\begin{align*}
\left[ dy_2 K_{11} W_{11} + dy_1 K_{12} W_{12} \right] \left[ dy_2 W_{11} + dy_1 W_{12} \right]^{-1}, & \quad \rho \in U_{11} \cap U_{12}, \\
\left[ dy_2 K_{21} W_{21} + dy_1 K_{22} W_{22} \right] \left[ dy_2 W_{21} + dy_1 W_{22} \right]^{-1}, & \quad \rho \in U_{21} \cap U_{22}, \\
\left[ dx_2 K_{11} W_{11} + dx_1 K_{21} W_{21} \right] \left[ dx_2 W_{11} + dx_1 W_{21} \right]^{-1}, & \quad \rho \in U_{11} \cap U_{21}, \\
\left[ dx_2 K_{12} W_{12} + dx_1 K_{22} W_{22} \right] \left[ dx_2 W_{12} + dx_1 W_{22} \right]^{-1}, & \quad \rho \in U_{12} \cap U_{22}.
\end{align*}
\]
Where,

\[
\begin{align*}
    dx_1 &= \frac{x - x_1}{x_2 - x_1}, & dx_2 &= \frac{x_2 - x}{x_2 - x_1}, \\
    dy_1 &= \frac{y - y_1}{y_2 - y_1}, & dy_2 &= \frac{y_2 - y}{y_2 - y_1}.
\end{align*}
\]

The final continuous controller can be formulated by considering all the possibilities of the position of scheduling variable within the block:

\[
k(\rho) = \begin{cases} 
    K_{11}, & \rho \in V_{11}, \\
    K_{21}, & \rho \in V_{21}, \\
    K_{12}, & \rho \in V_{12}, \\
    K_{22}, & \rho \in V_{22}, \\
    \bar{K}_1(\rho)W_1(\rho)^{-1}, & \rho \in U_{11} \cap U_{21} \cap U_{12} \cap U_{22}, \\
    \bar{K}_2(\rho)W_2(\rho)^{-1}, & \rho \in U_{11} \cap U_{12}, \\
    \bar{K}_3(\rho)W_3(\rho)^{-1}, & \rho \in U_{21} \cap U_{22}, \\
    \bar{K}_4(\rho)W_4(\rho)^{-1}, & \rho \in U_{11} \cap U_{21}, \\
    \bar{K}_5(\rho)W_5(\rho)^{-1}, & \rho \in U_{12} \cap U_{22}.
\end{cases} \tag{4-31}
\]

Where, \( V_{ij} \subset U_{ij} \) and it is defined as,

\[
\begin{align*}
    V_{11} &= [(U_{11} \cap U_{12}) \cup (U_{11} \cap U_{21}) \cup (U_{11} \cap U_{12} \cap U_{22})]' \\
    V_{21} &= [(U_{21} \cap U_{22}) \cup (U_{11} \cap U_{21}) \cup (U_{11} \cap U_{21} \cap U_{12} \cap U_{22})]' \\
    V_{12} &= [(U_{11} \cap U_{12}) \cup (U_{12} \cap U_{22}) \cup (U_{11} \cap U_{21} \cap U_{12} \cap U_{22})]' \\
    V_{22} &= [(U_{12} \cap U_{22}) \cup (U_{21} \cap U_{22}) \cup (U_{11} \cap U_{21} \cap U_{12} \cap U_{22})]' \\
\end{align*}
\]

Note that the allowable set of scheduling variable, \( \Lambda \), can be divided into several blocks and at any given time the scheduling variable can be placed in any one of the block and interpolation is performed in the aforementioned approach. In the literature of LPV systems, the stability analysis of a closed loop system involves the study of time variations of the scheduling variable \[33\]. Although the approach used in this thesis is conservative, we guarantee that stability can be preserved when the allowable parameter set satisfies the stability covering condition and when there is ‘no’ to ‘slow’ time variations in the scheduling variable.

### 4-2 Non-linear systems to Quasi-LPV systems

The horizontal plane model and vertical plane model explained in the previous chapter were put in the following form:

\[
\begin{align*}
    M\ddot{x}_1 &= f(x_1) + g(x_1, u_1), \tag{4-32} \\
    \dot{x}_2 &= h(x_1, x_2). \tag{4-33}
\end{align*}
\]

The state variables and control inputs for horizontal plane model and vertical plane model are summarized in Table 4-1.
The nonlinear dynamics of the system allows us to reformulate the given system into Quasi-LPV form. The assumptions and the substitutions that are involved in the reformulation are explained in this section. The resulting Quasi-LPV form is used as control plants for controlling the heading angle and vehicle depth.

### 4.2-1 Heading angle control plants (Quasi-LPV)

In the horizontal plane model, the longitudinal velocity \( u \) is the only term that introduces nonlinearity into the sway and yaw equations of motion. A Quasi-LPV form is obtained by exploiting this structure in the non-linear system. That is, the dynamical equations Eq. (3-59) and Eq. (3-60) are nonlinear with respect to the longitudinal velocity \( u \) and it can be rewritten in the following way:

\[
\begin{bmatrix}
  m - a_3 Y_f & -a_4 Y_r \\
  -a_4 N_f & I_z - a_5 N_r
\end{bmatrix}
\begin{bmatrix}
  \dot{v}(t) \\
  \dot{r}(t)
\end{bmatrix} =
\begin{bmatrix}
  (a_3 Y_f - m)u(t) \\
  (a_4 N_f)u(t)
\end{bmatrix}
\begin{bmatrix}
  v(t) \\
  r(t)
\end{bmatrix} +
\begin{bmatrix}
  a_2 Y_d, u^2(t) \\
  a_3 N_d, u^2(t)
\end{bmatrix} \delta_r(t).
\]

Therefore, augmenting \( \dot{\psi} = \dot{r} \) with the above equation and simplifying it, yields a Quasi-LPV system of the form:

\[
\begin{align*}
\dot{x}(t) &= A(\rho)x(t) + B(\rho)u_1(t), \\
y(t) &= Cx(t),
\end{align*}
\]

where,

\[
x = \begin{bmatrix} v & r & \psi \end{bmatrix}^T, u_i = \delta_r, y = \psi, \rho = u.
\]

It is important at this point to understand that the scheduling variable \( \rho \) is actually endogenous in the perspective of entire dynamics, which is, congruent to the previously explained definition of Quasi-LPV systems. Therefore, the steering control plants are obtained with heading angle as the system output and rudder angle as the system input.
4-2-2 Dive control plants (Quasi-LPV)

Similarly, in the vertical plane model, the pitch equation of motion Eq. (3-69) and kinematic equations Eq. (3-71), Eq. (3-72) are used in deriving the required Quasi-LPV control plant. The non-linearities in pitch motion Eq. (3-69) are due to the longitudinal velocity (u) and propeller rate (n). The simplified version of the control plant is given below:

\[
\begin{bmatrix}
\dot{q}(t) \\
\dot{\theta}(t) \\
\dot{z}(t)
\end{bmatrix} =
\begin{bmatrix}
a_1 u(t) + a_2 u(t)f(u(t), n(t)) & a_3 & 0 \\
1 & 0 & 0 \\
0 & -u(t) & 0
\end{bmatrix}
\begin{bmatrix}
q(t) \\
\theta(t) \\
z(t)
\end{bmatrix}
+ \begin{bmatrix}
b_1 u^2(t) + b_2 u^2(t)f(u(t), n(t)) \\
0 \\
0
\end{bmatrix}\delta_s(t),
\]

(4-36)

where, \(a_1, a_2, a_3, b_1, \) and \(b_2\) are constants. If we group the longitudinal velocity (u) and propeller rotation rate (n) as scheduling parameters, then we could write in Quasi-LPV form.

\[
\dot{x}(t) = A(\rho)x(t) + B(\rho)u_i(t),
\]

\[
y(t) = Cx(t),
\]

(4-37)

with,

\[
\rho = \begin{bmatrix} u \\ n \end{bmatrix}.
\]

Therefore, the dive control plants are obtained with vehicle depth as the system output and stern plane angle as the system input.

4-3 Heading angle control

In this section the autopilot design of heading angle is explained. Once the Non-linear model is reformulated into the Quasi-LPV form, the next steps are to design linear controllers and interpolation. The Quasi-LPV steering control plant is parameterized by a scalar scheduling variable, which is the longitudinal velocity (u). From the specifications of the ARIES AUV [28], it can be noted that the longitudinal velocity can vary from 0 to 1.8 m/s. Also, the rudder angle, which is the system input has a bound of \(\pm 22^\circ\). Five fixed points on the entire range of the surge velocity were chosen to cover the entire dynamics. The fixed points of the scheduling variables are:

\[
\rho = \begin{bmatrix} 0.10 & 0.52 & 0.95 & 1.50 & 1.80 \end{bmatrix}.
\]

(4-38)

The required performance specification is given in Table 4-3

4-3-1 Linear controllers

For each of the fixed scheduling parameter, the LTI plants are extracted and linear controllers are designed. The pole-zero map of the open loop plants are shown in Figure 4-4. It can
Table 4-2: Performance specification

<table>
<thead>
<tr>
<th>Description</th>
<th>Specification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Settling time</td>
<td>&lt;10 sec</td>
</tr>
<tr>
<td>Peak overshoot</td>
<td>&lt; 5%</td>
</tr>
<tr>
<td>Steady state error</td>
<td>0</td>
</tr>
</tbody>
</table>

Figure 4-4: Pole zero map of open loop system.

be noted that all the plants have a pole at the origin. A state feedback controller design based on LQR approach is used to get optimum reference tracking and to ensure stability at the fixed points of the scheduling variables (see section 4-1-3). The Q and R matrices of the cost function Eq. (4-17) are tuned to get desired response to a step reference of heading angle. The pole locations of the closed loop systems corresponding to the fixed scheduling variable are shown in Figure 4-5 and the responses to a step signal are shown in Figure 4-6. It can been seen that the higher the magnitude of the scheduling variable, the farther is the dominant pole from the origin. That is, the settling time for a step reference in heading improves with higher longitudinal velocities. For instance, the step response of the closed loop system corresponding to the fixed scheduling variable value of 0.1 has a settling time of 73 seconds and for 1.8 m/s the settling time is 5 seconds. It is quite logical to have obtained such a large settling time as the longitudinal velocity is almost zero. Recall that the control input (rudder angle) can be varied only up to $\pm 22^\circ$, indicating that the increase in feedback and feed forward gains will result in saturated control input after a point. Therefore, better controller performance is achieved for systems with higher values of longitudinal velocities. The systems obtained for longitudinal velocities of 1.8, 1.5, and 1 m/s meet the performance specifications.

The solution to the regulator equation exists, which is the precise condition for the existence
of a tracking controller. The solution of $\Pi$ and $\Gamma$ are:

$$\Pi = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \Gamma = 0.$$  \hspace{1cm} (4-39)
Solving for $G_i$,

$$ G_i = K_i \Pi = \begin{bmatrix} K_i(1) & K_i(2) & K_i(3) \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = K_i(3). \quad (4-40) $$

### 4-3-2 Interpolation

In order to implement stability preserving interpolation, the feedback gains must satisfy the stability covering condition (section 4-1-4). So far, the linear controllers are designed at fixed points of the scheduling variable. That is, $K_i$ renders $[A(\rho_i) - B(\rho_i)K_i]$ Hurwitz. In order to satisfy the stability covering condition the following steps are taken.

1. For every $\rho_i$, an open neighborhood $U_i$ is defined such that any system within the neighborhood is stabilized by the corresponding controller $K_i$. i.e, $[A(\rho) - B(\rho)K_i]$ is Hurwitz, $\forall \rho \in U_i$.

2. The neighborhood regions are allowed to intersect to its adjacent neighborhood to obtain a connected compact set. That is, $\Lambda \subseteq \bigcup_{i=1}^{\theta} U_i$.

Since the points in the scheduling variable are chosen to be equidistant, the neighborhood is defined to be at least more than half the distance between two given points. Then the stability of a single controller in the respective neighborhood is checked. In this case, the controller gain $K_i$ for the plant resulting from a $\rho_i$ is stabilizing the plant corresponding to $\rho_{i+1}$. This essentially means that a given neighborhood region can even cover the next point in the scheduling variable. Thus the stability covering condition is satisfied.

### 4-4 Depth control

In this section the autopilot design for the depth control is explained. The control plant for depth, which was previously brought into the Quasi-LPV form, consists of two scheduling parameters (longitudinal velocity and propeller speed) with stern plane angle as the control input and vehicle depth as the output. The longitudinal velocity can vary from 0 to 1.8 m/s and the propeller speed can vary from 0 to 157 rad/s. Five equidistant points in the scheduling parameters are chosen to form a grid and linear controllers are designed at the grid points. Therefore, 25 LTI systems are obtained and controllers are designed for each system.

$$ U = \begin{bmatrix} 0.10 & 0.52 & 0.95 & 1.37 & 1.80 \end{bmatrix} $$

$$ N = \begin{bmatrix} 0.00 & 39.25 & 78.50 & 117.75 & 157 \end{bmatrix} $$

The controller specification for a unit step signal is given as:

Anand Sundaresan

Master of Science Thesis
Table 4-3: Performance specification

<table>
<thead>
<tr>
<th>Description</th>
<th>Specification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Settling time</td>
<td>&lt;10 sec</td>
</tr>
<tr>
<td>Peak overshoot</td>
<td>&lt; 5%</td>
</tr>
<tr>
<td>Steady state error</td>
<td>0</td>
</tr>
</tbody>
</table>

4-4-1 Linear controllers

Linear controller design explained in section 4-1-3 is used to design controllers for 25 LTI systems (plants at the grid points). The open loop pole locations of all these plants are shown in Figure 4-7. Similar to the heading angle case, all the open loop plants have a pole at the origin. The Q and R matrices of the cost function Eq. (4-17) are tuned to get desired response to a step reference of heading angle. The pole locations of the closed loop systems corresponding to the fixed scheduling variable are shown in Figure 4-8. The step response of the closed loop systems are shown in Figure 4-9. The solution to the regulator equation exists, which is the precise condition for the existence of a tracking controller. The solution of $\Pi$ and $\Gamma$ are:

$$
\Pi = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \Gamma = 0. \tag{4-41}
$$

Solving for $G_i$,

$$
G_i = K_i \Pi = \begin{bmatrix} K_i(1) & K_i(2) & K_i(3) \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = K_i(3). \tag{4-42}
$$

4-4-2 Interpolation

The dynamics of the Quasi-LPV control plant for depth control is governed by two scheduling parameters. We are interested in the second case of interpolation given in section 4-1-4. In order to implement stability preserving interpolation, the feedback gains must satisfy the stability covering condition. The stability preserving interpolation is implemented in the following fashion:

1. At a given instant of time, the scheduling parameter is considered within a single block in the grid of allowable scheduling variable set.

2. For every $\rho_i$, an open neighborhood $U_i$ is defined such that any system within the neighborhood is stabilized by the corresponding controller $K_i$. i.e, $[A(\rho) - B(\rho)K_i]$ is Hurwitz, $\forall \rho \in U_i$.

3. The neighborhood regions are allowed to intersect to its adjacent neighborhood to obtain a connected compact set. That is, $\Lambda \subset \bigcup_{i=1}^{q} U_i$. 
4. Bilinear interpolation is used in the regions of intersection of all the four neighborhood regions.

5. Linear interpolation is used in the regions of intersection of two adjacent neighborhood regions.

Since the points in the scheduling variable are chosen to be equidistant, the neighborhood is defined to be at least more than half the distance between two given points. Then the stability of a single controller in the respective neighborhood is checked. In this case, the
controller gain $K_i$ for the plant resulting from a $\rho_i$ is stabilizing the plant corresponding to $\rho_{i+1}$. This essentially means that a given neighborhood region can even cover the next point in the scheduling variable. Thus the stability covering condition is satisfied.

### 4-4-3 Controller switching

The switching of controller is implemented in the following way:

1. Switching occurs only in the region of intersection of neighborhood regions.
2. The distance between the given point and the four vertices of the grid block is calculated.
3. The controller gain is switched to the gain at the point that is closest to the given point.

### 4-5 Illustrative example of speed control

Longitudinal velocity is the remaining parameter of interest to be controlled in the context of docking problem. A PI controller with anti-windup feature is implemented in the simulink model. In this design, no explicit model is used. The velocity signal from the model is compared with the reference signal to generate a error signal. The PI controller tries to maintain zero error at all instant. Taking saturation of control signal into consideration, anti-windup feature with back step method is used. It must be noted that the longitudinal
velocity and the propeller speed are the signals that schedules the heading angle and depth control plants.

4-6 Simulation results

Recall that the simulations involved in this thesis are based on the model of the ARIES AUV. It is a 6 degree of freedom model (Twelve equations of motion) in the form of matlab function obtained from [19]. The controller developed in the previous sections were tested in three different simulation setups:

1. 6 DOF, Non-linear model.
2. 3 DOF, Horizontal plane non-linear model.
3. 3 DOF, Vertical plane non-linear model.

Simulation setup In the previous chapter, it was shown that the 6 DOF dynamics can be decoupled into horizontal plane dynamics and vertical plane dynamics. The control plants were derived from these horizontal plane and vertical plane models by making further assumptions. The point that one needs to understand is that in the design procedure of the controller there have been several translations to scale down the abstract 6 DOF model. Therefore, we intend to the check the performance of the controller for both 6 DOF model as well as the 3 DOF models.

In the series of figures to follow, one can see the controller performance for heading angle, vehicle depth, and longitudinal velocity. The maximum command signal for longitudinal velocity (1.8 m/s) is given right from start of the simulation. Unit step reference signals for heading angle and depth are given after 50 seconds.

The comparison of 6 DOF and 3 DOF model response to a reference heading angle is shown in Figure 4-10. The response of the 3 DOF model is faster than the 6 DOF model. The controller performance for both the models are arguably satisfying the performance specification. Similarly, the response for a reference in vehicle depth is shown in Figure 4-11. There is not any significant difference in the controller performance for both the models. The surge velocity, which is the common signal in all the three simulink models is shown in Figure 4-12. One can notice a significant dip in the 6 DOF model. This dip is due the additional non-linearities present in the 6 DOF model. Also, it is to be noted that reference for heading angle and depth are given at same time instant. That is, we are deliberately exciting the horizontal plane dynamics and vertical plane dynamics at the same time, which can excite some of the modes of the 6 DOF model that is not present in the 3 DOF model. However, our design philosophy is to consider a spatially decoupled motion of the AUV. So, the motivation is to understand how the controllers would reacts when both the dynamics are excited simultaneously. In other words, we are investigating on the conservativeness of the controller, as in, does it push the non-linear system to instability? or how much is the controller performance affected? Finally, the trajectory of AUV is shown in Figure 4-13.
4-6 Simulation results

**Figure 4-10:** Heading angle

**Figure 4-11:** Depth
Figure 4-12: Comparison of Surge velocity

Figure 4-13: Propeller Speed

Figure 4-12: longitudinal Velocity

Figure 4-13: Trajectory of AUV for a unit step reference on heading angle and vehicle depth
Controllers were designed in Chapter 4 for controlling heading angle ($\psi$), vehicle depth ($Z$), and longitudinal velocity ($u$). A guidance algorithm intends to generate reference signals, which the controllers are required to track. The design of such guidance algorithms pertaining to docking problem are discussed in this chapter. It was mentioned in Chapter 1 that the scope of the thesis is low and medium level controllers. In that context, the reference generation is analogous to the medium level controller for the docking problem. The following aspects are covered in this chapter:

2. Translating the path into reference trajectories.

Recall from the problem statement defined in Chapter 2 that in this project, the approach for AUV motion control is spatially decoupled. For a docking scenario, the vehicle executes a maneuver in $xz$-plane and then in $xy$-plane. This essentially means that the reference signal for depth controller is the depth of the dock which can be assumed to be constant. Therefore, the following discussions focus on steering and velocity assignments.

### 5-1 Ingredients for guidance algorithms

#### 5-1-1 Way points tracking

**Objective:** Track way points $P_k$, $k = 0, 1, \ldots, n$.

In general, way point guidance systems are used to generate paths or trajectories for autonomous vehicles. The way points are stored in a database by a human operator, which
the vehicle uses to execute a certain maneuver. Note that, unlike target tracking, it is not important to exactly converge to a given way point $P_k$, but rather to use it as an aid to command the vehicle to traverse in a particular fashion.

**Heading angle reference ($\psi_{\text{ref}}$):** Pure pursuit guidance technique is a two point guidance technique: The interceptor and the target. The idea is it to align the interceptor’s velocity along the line of sight vector between the target and interceptor. Therefore, the angle subtended by the line of sight vector in global reference frame is given as the reference heading signal to the heading autopilot. Let $[x_a, y_a]$ and $[x_k, y_k]$ be the position of AUV and a way point in $xy$ – plane expressed in global coordinate frame. Then the reference heading angle can be computed by:

$$
\psi_{\text{ref}}(t) = \arctan2(y_k - y_a(t), x_k - x_a(t)),
$$

where, $\arctan2$ is four quadrant arctan. The idea is to track a given way point $P_k$ (using pure pursuit guidance) until the vehicle reaches a certain unit of distance close to the way point and then to track the next way point $P_{k+1}$. In [18] a circle of acceptance (2D) or sphere of acceptance (3D) criteria is used to implement such a switch in the way points.

$$(x_k - x_a(t))^2 + (y_k - y_a(t))^2 < c^2,$$

(5-1)

where, $c$ is the radius of the circle.

**Drawback:** One cannot guarantee that the vehicle will converge to the path between two way points. In fact, if the distance between two way points and the angle subtended by the paths are large, then one might witness a large cross track errors (distance from a given path). However, the docking problem can be seen as the final maneuver of a vehicle and the guidance technique must ensure that vehicle converges to the path which leads to the dock.

**5-1-2 Path following**

**Objective:** Converge to straight lines formed by way points.

Taking the draw back of the previous approach as the motivation, we are now interested in making the vehicle converge to a given path.

For $n$ way points, $n - 1$ straight line paths can be made. Let $P_k := [x_k, y_k]$ and $P_{k+1} := [x_{k+1}, y_{k+1}]$ be two consecutive way points forming a straight line and consider only the $xy$ coordinates of the AUV, as in $P_a := [x_a, y_a]$. Now, consider a path fixed reference frame with $P_k$ as origin. The frame is rotated by an angle $\psi_k$ (angle of the path) with respect to the global reference frame. Then the position of the vehicle with respect to the path fixed reference frame can be expressed as:

$$
\begin{bmatrix}
  s(t) \\
  e(t)
\end{bmatrix} =
\begin{bmatrix}
  \cos(\psi_k) & -\sin(\psi_k) \\
  \sin(\psi_k) & \cos(\psi_k)
\end{bmatrix}^T
\begin{bmatrix}
  x_a(t) - x_k \\
  y_a(t) - y_k
\end{bmatrix}.
$$

(5-2)
Where, \( s(t) \) is called as along track distance, \( e(t) \) is the cross track distance, and \( R^T(\psi_k) \) is the rotation matrix used for the coordinate transformation. In order to converge to the path, the \( \psi_{ref} \) assignment should ensure that cross track error is made zero \( (e(t) = 0) \). The two common approaches for \( \psi_{ref} \) assignments are: Enclosure based steering and Lookahead based steering. We will be using the lookahead based steering technique as it has more advantages compared to the former technique [36].

**Figure 5-1**: Lookahead Based Steering.

**Lookahead Based Steering**: The velocity vector of the vehicle is pointed towards the path at a lookahead distance \( (\Delta) \) from the projection of the vehicle position \( P_a \) onto the path. The geometry of this concept is shown in Figure 5-1. The steering assignment has two parts:

\[
\psi_{ref}(t) = \psi_k + \psi_r(t),
\]

where \( \psi_k \) is the angle of the path and \( \psi_r \) is the path velocity relative angle, which is given as:

\[
\psi_r(t) = \arctan \left( \frac{-e(t)}{\Delta} \right).
\]

Now, for \( n \) number of such paths we make use of the circle of acceptance criteria to switch the origin of the path fixed reference frame. That is, the origin of the frame is changed from \( P_k \) to \( P_{k+1} \) as the vehicle approaches \( P_{k+1} \).
5-2 Docking maneuvers

The focus of this section is to develop docking strategies based on the basic algorithms. From the discussions to follow, one can realize how powerful and effective, yet simple and intuitive are way points based path generation. The docking problem explained in Chapter 2 was based on time constraint. By using way points and conventional line of sight guidance approach we can translate it into geometrical path following problems.

Obviously the docking strategies aim in tackling the docking problem explained in Chapter 2. Therefore, the docking problem is presented once again so that the reader need not switch back and forth from Chapter 2 and this Chapter.

\[ P_a(t) = P_d, \quad t > T_{d1}, \]  
\[ u_a(t) = u_d, \quad t > T_{d1}, \]  
\[ \psi_a(t) = \psi_d, \quad t > T_{d2}, \]  
\[ z_a(t) = z_d, \quad t > T_{d3}, \]  
\[ 0 < T_{d3} < T_{d2} < T_{d1}. \]  

5-2-1 Three point docking maneuver

**Objective:** Given the initial position and orientation of AUV \((P_{ai})\) and dock \((P_{di}, \psi_{di})\), generate three way points / two piecewise paths. The two paths are:

1. **Terminal path:** The path leading to dock.
2. **Guide path:** The path that is used to guide the AUV to the terminal path.

The problem reduces to cross track error control. Parameters that can be used to tune the vehicle trajectory: Lookahead distance \((\Delta)\) and circle of acceptance \((c)\)

**Path generation**

Let the three way points be defined as:

\[ P_{k1} = \begin{bmatrix} x_{k1} & y_{k1} & z_{k1} \end{bmatrix}, \]
\[ P_{k2} = \begin{bmatrix} x_{k2} & y_{k2} & z_{k2} \end{bmatrix}, \]
\[ P_{k3} = \begin{bmatrix} x_{k3} & y_{k3} & z_{k3} \end{bmatrix}. \]

The straight line joining \(P_{k1}\) and \(P_{k2}\) is termed as **Guide path** and the straight line joining \(P_{k2}\) and \(P_{k3}\) is termed as **Terminal path**.

The three way points are generated in the following fashion:

Anand Sundaresan Master of Science Thesis
1. Last way point is the position of the dock itself. \( P_{k3} = P_d = [x_d, y_d, z_d] \).

2. All the three way point are in same depth as the dock. Meaning \( z_{k1} = z_{k2} = z_{k3} = z_d \).

3. The penultimate way point must be \( d \) distance (distance to dock) behind the dock and along the x-axis of the dock’s body reference frame.

4. First way point is created at \( d_1 \) distance from the penultimate way point, by looking along the line of sight vector from the penultimate way point to the AUV position.

Since all the way points are in same depth, we can now focus on the \( xy\)-plane alone. Consider only the \( x \) and \( y \) terms in \( P_{k1}, P_{k2}, \) and \( P_{ai} \). The penultimate way point can generated as:

\[
P_{k2} = P_d + R(\psi_d) \begin{bmatrix} -d_0 \ 0 \end{bmatrix}^T.
\]

Once the penultimate way point is generated, we fix a frame at \( P_{k2} \) and obtain the initial position of the AUV in this frame and calculate the line of sight angle from the origin (which is \( P_{k2} \)):

\[
P_{atemp} = R^T(\psi_d)(P_{ai} - P_{k2}),
\]

\[
los = \arctan 2(P_{atemp}(2), P_{atemp}(1)).
\]
Using the los angle the first way point can be calculated as:

\[ P_{k_1} = \begin{bmatrix} d_1 \cos(\text{los}) \\ d_1 \sin(\text{los}) \end{bmatrix}. \]

The concept is shown in Figure 5-2. The way we generate these way points are arguably similar to inverse kinematics, wherein based on the end effector’s position the joint positions are calculated. Here given the position and angle of the dock, we find the positions that satisfy a particular set of conditions.

Once the way points are generated, the path following approach is used to follow the path.

**Path following**

Assuming that the tracking controllers can satisfy,

\[ u_a(t) - u_{ref}(t) = 0, t \to \infty, \quad (5-8) \]
\[ z_a(t) - z_{ref}(t) = 0, t \to \infty, \quad (5-9) \]
\[ \psi_a(t) - \psi_{ref}(t) = 0, t \to \infty. \quad (5-10) \]

The three point docking strategy tackles the docking problem in the following ways:

1. Converging to the Guide path will satisfy (but not necessarily guarantee) depth constraint Eq. (5-6), as all the way points are in same depth as that of dock.

2. Converging to the terminal path will satisfy (but not necessarily guarantee) angle constraint Eq. (5-5). Furthermore, since the path is aligned in same orientation as that of dock and by having a non-zero command signal \( u_{ref} \) for surge velocity it will lead to the dock satisfying Eq. (5-3).

3. Finally a desired surge velocity can be commanded as the vehicle is within the circle of acceptance to satisfy Eq. (5-4).

Note that the time constraint Eq. (5-7) is automatically satisfied as the problem is translated to geometrical paths.

**Cross track error control** Once the vehicle and dock are in same depth, the performance evaluation of path following boils down to the cross track error \( e(t) \). Ideally, one would want local exponential convergence of the cross track error to zero. By specifying "local" we confine the discussion to a single piecewise path. By knowing the decay rate of error one can decide on the minimum distances \( d \) and \( d_1 \) to extend the argument on guaranteeing docking.
5-2 Docking maneuvers

5-2-2 N-point docking maneuver

Objective: Given the initial position and orientation of AUV \((P_{ai})\) and dock \((P_{di}, \psi_{di})\), generate \(N\) way points / \((N-1)\) piecewise paths. The paths are:

1. Terminal path: The path leading to dock.

2. Intermediate paths: \(N-2\) paths are generated until the line of sight angle from the way point to AUV satisfies a condition.

The problem reduces to cross track error control. Parameters that can be used to tune the vehicle trajectory: Lookahead distance \((\Delta)\), circle of acceptance \((c)\), and the angle constraint \((\psi_k)\).

This approach is basically an extension to the Three point maneuver. The idea is to say that the angle subtended by two consecutive paths should be more than a minimum value. The motivation for such an idea is to make a smoother transition from Guide path to Terminal path. This approach allows one to have one more tuning parameter for tuning the vehicle’s trajectory.

5-2-3 Docking maneuver for non-stationary dock

Objective: Generate new paths (on-line), if the position of dock changes during the docking maneuver.

This scenario is interesting mainly because it resembles the pipelay scenario of Allseas. That is, the pipelay vessel moves a constant distance after laying every segment of the pipe. It can be assumed that the dock is connected to the pipelay vessel and moves along with the pipelay vessel. Therefore, we are interested in giving a step change in the position of the dock during the docking maneuver and still manage to dock successfully.

The autopilot and guidance systems are present in the AUV, whereas the position of the dock is communicated to the AUV from the dock. Consequently, it is logical to assume that the position update of the dock is much slower than that of the position update of the AUV. Moreover, underwater communications are considerably slow when compared to air or land. Therefore, we are only considering a static change in the position rather than a dynamic change in the position.

The problem is tackled in the following way:

1. Till there is an update in the dock position, the initial positions of the AUV \((P_{ai})\) and dock \((P_{di}, \psi_{di})\) is used for the way point generation.

2. Once there is an update in the dock position, the guidance system discards all the future waypoints and generates new points based on the new position of the dock and the position of AUV at the instant of update from dock’s position.
5-3 Simulation results

For the simulation results to follow, we use the 6 DOF non-linear model of the ARIES AUV.

5-3-1 Path following

From the discussions on docking strategies one can note that the common denominator in all those strategies is path following. It would be interesting to discuss about the effect of lookahead distance ($\Delta$) on vehicle trajectory before presenting the results of the docking approaches. In Figure 5-3 the trajectory of AUV with initial position [90 50 0] and initial heading angle of 0° is shown. The vehicle intends to converge to the straight line formed by two way points: wp1 = [100 100 1] and wp2 = [200 200 1].

Two important facts:

1. Oscillatory behavior is seen for low values of $\Delta$.
2. Over damped behavior is seen for high values of $\Delta$.

Recall that the steering assignment involves the following equations:

$$\psi_{ref}(t) = \psi_k + \psi_r(t), \quad (5-11)$$

with,

$$\psi_r(t) = \arctan\left(\frac{-e(t)}{\Delta}\right). \quad (5-12)$$

The constant term $\psi_k$ in Eq. (5-11) is the angle of a given path with respect to the global reference frame, while the $\psi_r(t)$ is the corrective term responsible for convergence. At each time instant $\psi_r(t)$ is such that the error becomes zero at $\Delta$ distance ahead in the path reference frame (see Figure 5-1 for diagram).

If Eq. (5-11) consists only the path angle $\psi_k$, then one can expect the vehicle trajectory to be exactly parallel to the path. Hence it is the $\psi_r(t)$ (corrective term) which makes the vehicle converge to the path. This approach is very much similar to an model predictive control approach. That is, in MPC, at time instant $t$ the controller uses a model to predict future states within a prediction horizon (T) and optimizes a cost function to find the best control input to be given to the plant at time $t$. At the next time step the entire process happens again. Here, the $\psi_r(t)$ term generates an angle (like a control input) at each time step so that the error becomes zero at lookahead distance ($\Delta$).

If this approach can be called as "similar" to MPC approach then it is equally important to understand the fundamental difference. In MPC, the controller uses the future state trajectory and future inputs from time instant $t$ to $t + T$ in a cost function and solves for the best input. However, here it is equivalent of saying that the controller is interested in one future instant which is $\Delta$ distance ahead. Therefore, the controller cannot see if the future trajectory will overshoot the reference trajectory or not and so one needs tune the lookahead distance to get the desired trajectory.
Figure 5-3: Effect of lookahead distance ($\Delta$).
5-3-2 Case one: Stationary docking station

In case one, the position $P_d$ and orientation of dock $\psi_d$ is constant throughout the simulation. The simulation is setup in the following way:

- AUV initial position: $P_{ai} = [100 \ 100 \ 0]$.
- AUV initial orientation: $\psi_{ai} = 0$.
- Dock initial position: $P_{di} = [300 \ 200 \ 5]$.
- Dock initial orientation: $\psi_{di} = 0$, $\psi_{di} = \pi/4$, $\psi_{di} = \pi/2$, $\psi_{di} = \pi$

Three point docking maneuver

The trajectory of AUV in $xy$- plane executing the docking maneuver for four different orientations of dock is shown in Figure 5-4 and the respective cross track errors are show in Figure 5-6. The desired results are obtained with the radius of circle of acceptance as $c = 20$ meters and a lookahead distance of $\Delta = 15$ meters. For $\psi_{di} = 0$ and $\psi_{di} = \pi/4$, the simulation yields desired results. However, for $\psi_{di} = \pi/2$ and $\psi_{di} = \pi$ one can witness a overshoot when the vehicle transits from guide path to terminal path. The cross track error $e$ overshoots up to 20 meter in this situation (see Figure 5-6).

This is the major drawbacks of this approach. The purpose of having the guide path was to facilitate quick convergence to terminal path, which seems not so useful now. Of course, one might argue that the difference in orientation of AUV and Dock is so huge that such a result is expected. Also, for greater difference in orientation the circle of acceptance ($c$) can be increased to avoid overshoot but such a change will cost at slow convergence to terminal path. The motivation for N-point docking maneuver was this specific drawback.

N- point docking maneuver

The trajectory of AUV in $xy$- plane executing the N point docking maneuver for four different orientations of dock is shown in Figure 5-5 and the respective cross track errors are show in Figure 5-7.

N point docking maneuver overcomes the drawback of previous approach at the cost of increasing the distance to dock and eventually increasing the time to dock. This approach allows one more parameter, path angle ($\psi_k$) to tune the vehicle trajectory. By using the circle of acceptance ($c$) and path angle ($\psi_k$) we could impose bounds on cross track error ($e$) while transitioning from one path to another. Locally, as in, in a single piecewise path, such a bound gives us the confidence that the initial cross track error will be within the bound ($\bar{e}$). This is $\bar{e}$ will translate into bounds for the commanded reference heading angles ($\bar{\psi}_{ref}$), which is given as:

$$\bar{\psi}_{ref} = \psi_k + \arctan \left( \frac{-\bar{e}}{\Delta} \right)$$ (5-13)
a cross track error bound ($\bar{e}$) of ±6 meters is achieved for 1.5 m/s vehicle speed by tuning the parameters as:

$$c = 15 \text{ m}$$
$$\Delta = 15 \text{ m}$$
$$\psi_k \geq 2.6 \text{ rad}$$

In Figure 5-8, the controller performance for Three point and N-point docking strategies are compared. The figure is shown for $\psi_{di} = \pi$, which had the maximum overshoot in cross track error. One can see the effect of bound on the controller performance. In N-point strategy it is as if the reference signal is increased in small steps while in Three point strategy it happens in one step. Therefore the N-point strategy does not command unrealistic reference signals to the controller.

5-3-3 Case two: Non-stationary docking station

In case two, a step change in the position $P_d$ and orientation of dock $\psi_d$ is investigated. The simulation is setup in the following way:

- AUV initial position: $P_{ai} = [100 \quad 100 \quad 0]$.
- AUV initial orientation: $\psi_{ai} = 0$.
- Dock position at $t = 0$: $P_{di} = [300 \quad 200 \quad 5]$.
- Dock position at $t = 100s$: $P_{di} = [400 \quad 200 \quad 5]$.
- Dock orientation at $t = 0$: $\psi_d = 0$.
- Dock orientation $t = 100s$: $\psi_d = 0$, $\psi_d = \pi/4$, $\psi_d = \pi/2$, $\psi_d = \pi$.
Figure 5-4: Three point docking maneuver.
Figure 5-5: N-point docking maneuver.
Figure 5-6: Three point docking maneuver - Cross track error.
Figure 5-7: N-point docking maneuver - Cross track error.
Figure 5-8: Comparison of heading controllers in closed loop for $\psi_d = \pi$. 

Anand Sundaresan Master of Science Thesis
Figure 5-9: Three point docking maneuver for non-stationary dock.
Figure 5-10: N-point docking maneuver for non-stationary dock.
Figure 5-11: Three point docking maneuver for non-stationary dock - Cross track error.
Figure 5-12: N-point docking maneuver for non-stationary dock - Cross track error.
Conclusions and Recommendations

6-1 Conclusions

In the beginning of this thesis report, a docking problem was formulated. The problem was approached in the following sequence:

- **Variable of interest**: The first step was to choose a set of variables to control for effective motion control. The vehicle heading angle $\psi_a$, vehicle depth $z_a$, and the longitudinal velocity $u_a$ were chosen so that the problem can be confined to spatially decoupled motions.

- **AUV’s equations of motion**: We investigated on modeling of underwater vehicles to obtain the equations of motion. The 6 DOF model primarily consists of two matrix equations: dynamics and kinematics. The matrix form of equations were translated to component form resulting in individual equations for surge, sway, yaw, roll, pitch, and yaw with the respective kinematic equations. Once the model was translated to component form we decoupled it to horizontal plane (3 DOF) and vertical plane models (3 DOF), which were used in control design. For simulations purposes we used the parameters of ARIES AUV from the literature.

- **Controller design**: A gain scheduled state feedback and feed forward controllers were designed in LPV framework for controlling heading angle and depth. Third order Quasi-LPV control plants with one dimensional and two dimensional scheduling variables were derived for heading angle and depth respectively. Gains were designed for frozen values of scheduling variables using regulator equation. Since the design is reference tracking controller, the output error was added as the fourth state to have integral action. Consider slow time variations in scheduling parameters, a stability preserving interpolation of gains were implemented to achieve the final controller. A PI controller with anti windup feature was implemented for longitudinal velocity control. Finally, the controller was tested on both 3 DOF and 6 DOF models of ARIES AUV.
• Guidance strategies: Two docking strategies: Three point and N-point, were developed using lookahead based path following and investigated their performance for two scenarios: stationary dock and non-stationary dock. The docking strategies addressed the docking problem in two aspects: geometric path generation and path following. It was shown that the N-point docking strategy yielded better convergence to the paths than the three point docking. Finally, the heading controller performance in closed loop was compared for the two developed strategies.

6-2 Recommendations

Now that the conclusion of this work is given, it is interesting to discuss about alternate approaches to the questions that were formulated in the beginning. In the following discussion, we intend to explain alternate approaches and discuss its impact on certain decisions.

• Different choice of variables to control: As mentioned several times, the spatially decoupled approach was chosen for AUV motion control in 3D, which required the control of heading angle, longitudinal velocity and vehicle depth. Recall that there was a single assumption which allowed us to use this approach: The dock has no pitch angle. If that is not the case, the problem would extend to spatially coupled motion control, which would require us to control pitch angle rather than depth. Moreover, accommodating for dock’s pitch angle is more realistic than assuming that it would be constant. If pitch angle needs to be controlled then the following changes must be done. Recall that the linear controllers for gain scheduled depth control was designed using the regulator equation. It was shown that the solution to the regulator equation was \( \Pi = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T \) and \( \Gamma = 0 \). If we choose to control the pitch angle \( \theta_a \), the solution changes which is shown below:

The state vector for depth control plant was \( x = \begin{bmatrix} q & \theta & z \end{bmatrix}^T \). The C matrix will change from \( \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \) to \( \begin{bmatrix} 0 & 1 & 0 \end{bmatrix} \).

The regulator equation:

\[
\begin{pmatrix}
A & B \\
C & 0
\end{pmatrix}
\begin{bmatrix}
\Pi \\
\Gamma
\end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}.
\]

The new solutions of \( \Pi \) and \( \Gamma \) would be:

\[
\Pi = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \Gamma = \frac{A(1,2)}{B(1)}.
\]

where, \( A(1,2) \) is a constant term in the \( A(\rho) \) matrix of Quasi-LPV plant, where as \( B(1) \) is a function of scheduling variable. In short, the feed forwards gain will become:

\[
G_i = \Gamma(\rho_i) + K_i \Pi. \quad (6-1)
\]
• **3D Path following:** As an extension of the previous point, control of vehicle pitch would allow us to design 3D path following controllers. In 2D path following, using the 2D rotation matrix, we could obtain the vehicle’s position in a path fixed reference frame as: Along track error \( s \) and cross track error \( e \). In 3D, the track errors will be \( \begin{bmatrix} s & e & h \end{bmatrix}^T \), where \( h \) is the *vertical track error* [36]. Therefore, a track error controller would generate reference signals for heading angle, pitch angle, and longitudinal velocity simultaneously and hence a spatially coupled maneuver would track the path in 3D.

• **Path generation as optimization problem:** Perhaps, one might notice that the docking problem was approached only with conventional guidance strategies. Although it might look straightforward to use only optimization instead of conventional geometrical approach, one must not overlook the reality. That is, using only optimization to generate reference signals as suggested in [37] is to say that we are generating an optimized path of, for example, 100 meters of length. A wiser way would be to blend conventional strategies with optimization. In [38], the lookahead distance \( \Delta \) is considered as a time varying signal and varied using MPC. Such a strategy allows them to simultaneously obtain minimum convergence time to path and minimum overshoot. In [39], a line of sight based MPC is used for way point tracking. Therefore, the most interesting areas to investigate would be to club these path following strategies for docking scenarios.

• **Non-linear controller:** In gain scheduling, the design of linear controllers at fixed scheduling variables is laborious and tedious task. Moreover, in spite of making such efforts in linear controllers, guaranteeing exponential tracking of controller is also not easy. If 3D path following is to be done, a non-linear controller is a better choice in both stability as well as performance aspects [40] [41] [42].
Bibliography


Anand Sundaresan Master of Science Thesis


