FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

# Autonomous Surface Vehicle based docking for an Autonomous Underwater Vehicle

**Miguel da Fonseca Rosa**

Mestrado Integrado em Engenharia Eletrotécnica e de Computadores

Supervisor: João Borges de Sousa (Eng.)

June 2017

# Resumo

Veículos autónomos subaquáticos tem sido amplamente usado como ferramentas em operações subaquáticas. Autonomia significa a capacidade de um planeamento deliberativo integrado sem qualquer interação humana. Os níveis de autonomia de veículos subaquáticos têm vindo a aumentar ao longo dos anos. Um facilitador essencial para a operação autónoma tem sido o *docking*. *Docking* autónomo irá permitir que os veículos possam operar independentemente da intervenção humana, em regiões remotas. O conceito de *docking* vai desde estações de *docking* subaquáticas até ao lançamento e resgate através de embarcações de superfície autónomas.

O Laboratório de Sistemas e Tecnologias Subaquáticas (LSTS) tem vindo a desenvolver e usar este tipo de veículos há mais de vinte anos e por isso surgiu a necessidade de desenvolver uma forma de *docking*.

Esta dissertação pretende desenvolver uma abordagem ao *docking* entre dois veículos existentes na frota do LSTS. Estes veículos seriam o Veiculo Autónomo de Superfície (ASV) *Caravela* e um Veiculo Autónomo Subaquático (LAUV).

Por esta razão, foi estudado e discutido a derivação e simplificação de equações de movimento para veículos de superfície e subaquáticos e conhecidas as abordagens ao *docking*, entre uma doca fixa ou em movimento.

Foi estudado uma forma de obter a localização entre os dois veículos, recorrendo ao Sistema de Posicionamento Global (GPS) e/ou *Ultra-Short Baseline* (USBL) e a Unidade de Medida Inercial (IMU) atualmente disponíveis nos sistemas.

Várias máquinas de estado foram implementadas para que houvesse uma manobra de *docking* otimizada e à prova de falhas, mesmo quando existe perturbações externas, tais como correntes marinhas sobre a abordagem de *docking*.

Toda a abordagem foi implementada usando a *toolchain* desenvolvida no LSTS (DUNE, IMC, Neptus).

# Abstract

Autonomous Underwater Vehicles have been used as tools for underwater operations. Autonomy means the capability to perform deliberative planning on-board without any human interaction. The levels of autonomy of underwater vehicles have been increasing over the years. One essential enabler for autonomous operation concerns docking. Autonomous docking will allow vehicles to operate independently of human interaction in remote areas. Docking concepts range from underwater docking stations to launch and recovery from autonomous surface vessels.

The Underwater Systems and Technologies Laboratory (LSTS) has been developing and using this type of vehicles for more than twenty years and therefore the need of automatic docking of vehicles has emerged.

This dissertation aims to develop an approach for docking between two existing vehicles from the LSTS fleet. The vehicles would be an Autonomous Surface Vehicle (ASV) *Caravela* and a Light Autonomous Underwater Vehicle (LAUV).

For this reason, we study and discuss the derivation and simplification of motion equations for underwater and surface vehicles and known approaches for docking, between a fixed or a moving dock.

We studied a way to have localization between the two vehicles using Global Positioning System (GPS) and/or Ultra-Short Baseline (USBL) and the internal Inertial Measurement Unit (IMU), currently available on the systems.

Several state machines where implemented in order to have an optimized and fail-safe docking maneuver even having external disturbances like sea-currents on the docking approach.

The whole approach was implemented using the tool-chain developed in LSTS (DUNE, IMC and Neptus).

# Acknowledgements

First, I would like to thank to my supervisor João Borges de Sousa for the opportunity to develop this project under his supervision, giving me guidance and advices that led me to accomplish this dissertation.

I would like to thank my lab partner João Teixeira, which helped me understand some of the most difficult subjects with simples explanations and had the "patience" to do it.

I would like to thank also to my lab partners Pedro Gonçalves and Maria Costa for having the time to help me with the implementation of *DUNE* and José Pinto and Paulo Dias for the help in making my maneuver in the *Neptus* software.

I also have to note the tremendous help and support given my girlfriend Mariana Lopes in the making of this dissertation.

Miguel Rosa

*“The right man in the wrong place can make all the difference...*
*in the world.”*

The G-Man, Half-Life 2

# Contents

# List of Figures

# List of Tables

# Acronyms

| | |
|---|---|
| AHRS | Attitude and Heading Reference System |
| AUV | Autonomous Underwater Vehicle |
| ALIVE | Autonomous Light Intervention Vehicle |
| ASV | Autonomous Surface Vehicle |
| ADS | Autonomous Docking Systems |
| CG | Center of Gravity |
| CB | Center of Buoyancy |
| DOF | Degrees of Freedom |
| DUNE | DUNE Unified Navigation Environment |
| DP | Dynamic Positioning |
| ECEF | Earth-Centered Earth-Fixed coordinate system |
| EM | Electromagnetic |
| ESC | Electronic Speed Controller |
| EOM | Equations of Motion |
| FSM | Finite-State Machine |
| LAUV | Light Autonomous Underwater Vehicle |
| NED | North-East-Down coordinate system |
| PID | Proportional-Integral-Derivative |
| SBC | Single Board Computer |
| SISO | Single-Input Single-Output |
| SNAME | Society of Naval Architects and Marine Engineers |
| IMU | Inertial Measurement Unit |
| IMC | Inter-Module Communication |
| I-AUV | Intervention AUV |
| INS | Inertial Navigation System |
| TOA | Time of Arrival |

# Chapter 1

# Introduction

## 1.1 Motivation

The planet Earth is covered by a large body of water, whether is oceans, lakes or rivers. This water mass has a great influence on ecosystems and therefore hugely important to study it. Also the problem of climate change has been a subject that has impulsed the studies of the Earth's body water masses. Hence, its necessary to build the essential tools for the research of technological and scientific approaches in exploration and analysis of aquatic environments.

One of the most known technological breakthrough in aquatic exploration is the Autonomous Underwater Vehicle which is a torpedo-like shaped vehicle with the capability of being operated remote and independently.

Since the vehicles have internal power and wide range of scientific sensors, such as sonar for sweeping the ocean floor or water analysis sensor, to analyze the values of pH and Rhodamine in water samples. Being autonomous also means that they can work for long periods of time without human intervention with a predefined plan.

Ocean exploration has been the motivation for the Underwater Systems and Technologies Laboratory (LSTS) to develop a great variety of autonomous vehicles.

Some of this systems are Autonomous Surface Vehicles (ASV), Autonomous Underwater Vehicles (AUV) and Remotely Operated Vehicles.

Autonomous Surface Vehicle are a catamaran-based shape vehicles capable of have a huge load of sensors and actuators and are extremely capable of moving on the surface of the water.

Figure 1.1: A typical Autonomous Surface Vehicle. Source - LSTS

Autonomous Underwater Vehicles are a powerful tool in underwater data gathering. They have a torpedo shape and operate independently from the surface, having no physical link. This vehicle are capable of having a huge payload under a small package and are capable of making a mission with different objectives, simultaneously.


Figure 1.2: A typical Autonomous Underwater Vehicle. Source - LSTS

Remotely Operated Vehicles are submersibles physically connected to the surface by an umbilical cable that provides power and communications. They are specialized vehicles capable of perform many operations such as operate in hazardous environments or underwater inspections.

Figure 1.3: A typical Remotely Operated Vehicle. Source - LSTS

The LSTS has contributed to a positive evolution of knowledge and technology by having a structured ocean exploration approach using multiple vehicles simultaneously. Their vision is to implement a network multi-vehicle operation consisting in a number of underwater vehicles, supported by other systems and platforms[1].

This dissertation intents to have a docking approach between two systems of the LSTS fleet in order to perform an autonomous retrieve of the vehicle and data/power transfer over two systems.

## 1.2 Related Work

Docking of two autonomous systems has been widely published in literature with the motivation being the ability to dock two vehicles for recharging, data upload and inspection [Singh *et.al* [2]]. There are some examples of docking systems that use acoustic homing to obtain range and heading from the target, as described by Stokey *et. al.* [3] and McEwen *et. al.* [4]. Similar approaches were applied by Freezor *et. al.* [5] using electromagnetic guidance. Using different types of sensors combined, docking with vision and acoustics was proposed by Evans *et. al.* [6] where acoustic homing is used until the vehicle gets within the visual range of the target and uses visual identification for the final stage of the docking. A docking approach where the docking station is movable was proposed by Braga [7] where it was implemented a hierarchical control approach for the docking between a ROV and an AUV.

## 1.3 Objectives

This dissertation intents to create a docking approach using the LSTS fleet and toolchain. The vehicles would be an Autonomous Surface Vehicle (ASV) named "Caravela" and a Light Autonomous Underwater Vehicle (LAUV) from the lab fleet. The main objective is to develop an effective approach to the docking problem, with two systems that are not intended to. A maneuver should be implemented and controllers should be built in order to have the vehicles performing the approach with success. There should be some way to communicate between the vehicles and the interface that plans the maneuver. A collision avoidance between the two vehicles must be applied in order to not destroy the vehicles while attempting the maneuver. All the tests must be simulated first using the toolchain capabilities for such.

## 1.4 Outline

- Chapter 1 introduces the motivation and objectives of this dissertation.
- Chapter 2 shows the background material required to the development of this dissertation.
- Chapter 3 presents the current state of the art of autonomous docking system for underwater vehicles.
- Chapter 4 declares the existing problem associated with autonomous docking.
- Chapter 5 demonstrates the *problem approach* to solve the described problems.
- Chapter 6 displays the results of the *practical approach* of the problem.
- Chapter 7 justifies the conclusions of this work and includes some guidelines for future work regarding autonomous docking systems.

# Chapter 2

# Background

This chapter introduces the background required to the development of this dissertation. Models of Marine Vehicles are presented and studied. Hybrid systems theory is introduced and an application example is shown. An overview of the LSTS toolchain is displayed.

## 2.1 Models of Marine Vehicles

This section elaborates the mathematical approach of the AUV and ASV kinematics and dynamic models. It follows Fossen [8] who presented the six degrees of freedom (DOF) equations of motion (EOM) for marine vehicles.

### 2.1.1 Kinematics

In order to determine the position and orientation of a marine vehicle in a given location, six coordinates are necessary. The first three coordinates are related with the position and translational motion along x-, y- and z axes. The last three coordinates deal with the attitude and angular rates. In marine vehicles, the six different motion components are typically defined as: *surge, sway, heave, roll, pitch* and *yaw*.

#### 2.1.1.1 Coordinate frames

When conceptualizing the motion of marine vehicles in 6 DOF it's convenient to define two coordinate frame as indicate in 2.1.

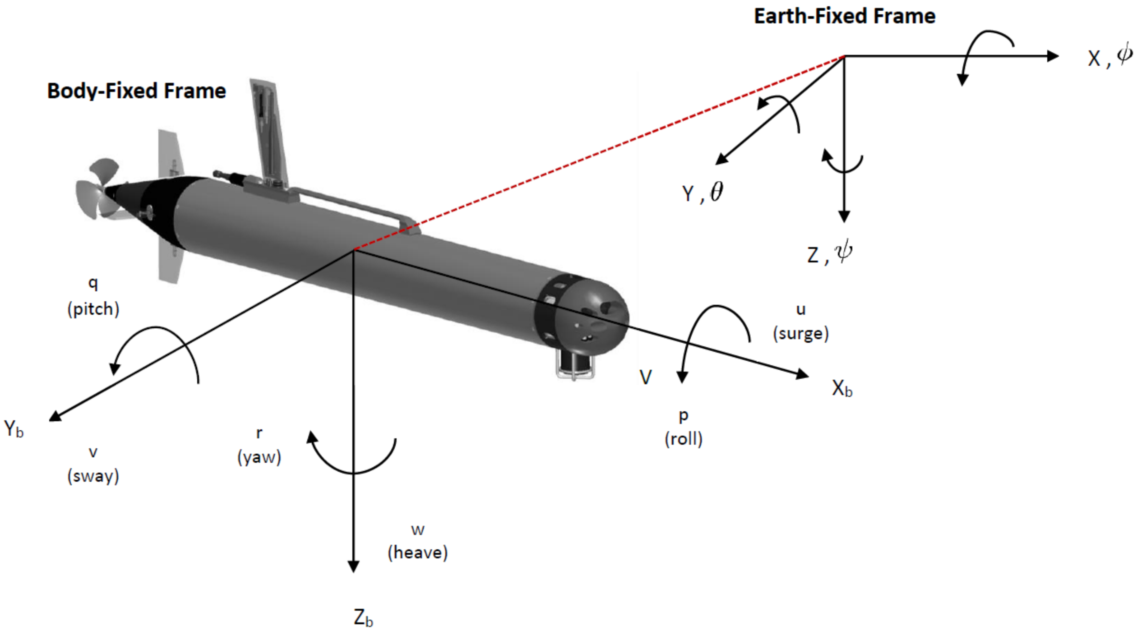


Figure 2.1: Coordinate frames between body-fixed and earth-fixed

The moving coordinate frame $X_b$,$Y_b$,$Z_b$ is fixed to the vehicle and is called the body-fixed reference frame. This frame is described to an inertial reference frame and for marine vehicles it's usually assumed that the accelerations of a point on the surface of the Earth hardly affects low speed vehicles.

| DOF | | Forces and Moments | Linear and angular velocity | Position and Euler angles |
|---|---|---|---|---|
| 1 | Motion in the x-direction (surge) | $X$ | u | x |
| 2 | Motion in the y-direction (sway) | Y | v | y |
| 3 | Motion in the z-direction (heave) | Z | w | z |
| 4 | Rotation about the x-axis (roll) | K | p | $\phi$ |
| 5 | Rotation about the y-axis (pitch) | M | q | $\theta$ |
| 6 | Rotation about the z-axis (yaw) | N | r | $\psi$ |

Table 2.1: Notation used for marine vehicles

Has seen in figure 2.1, we usually have two reference frames, where W-frame is coupled to the world, the x-axis points to north, the y-axis to the east and z-axis points forward to the center of the Earth. The B-frame is coupled to the body of the vehicle and the x-axis points to the forward direction, the y-axis to the right of the vehicle and the z-axis vertically down. Given this, all the DOF for the B-frame and W-frame can be listed in a vector form in

$$\chi = \begin{bmatrix} surge & sway & heave & roll & pitch & yaw \end{bmatrix}^T = \begin{bmatrix} x_B & y_B & z_B & \phi_B & \theta_B & \psi_B \end{bmatrix}^T \tag{2.1}$$

and

$$\eta = \begin{bmatrix} x_W & y_W & z_W & \varphi_W & \theta_W & \psi_W \end{bmatrix} \tag{2.2}$$

#### 2.1.1.2 Euler Angles

This section presents the equations that associate the body-fixed reference frame to the earth-fixed reference frame for a vehicle of 6 DOF. If defined:

$$\begin{aligned} \eta &= [\eta_1^T, \eta_2^T] \quad where \quad \eta_1 = [x, y, z] \quad \eta_2 = [\phi, \theta, \psi] \\ \nu &= [\nu_1^T, \nu_2^T] \quad where \quad \nu_1 = [u, v, w] \quad \nu_2 = [p, q, r] \end{aligned} \tag{2.3}$$

where $\eta$ is the position and orientation of the vehicle expressed in the earth-fixed reference frame and $\nu$ is the linear and angular velocities of the vehicle in the body-fixed reference frame. The relation between $\eta$ and $\nu$ if:

$$\dot{\eta}_1 = J_1(\eta_2)\nu_1 \tag{2.4}$$

where $J_1(\eta_2)$ is a transformation matrix which is related through the functions of the Euler angles: roll $(\phi)$, pitch $(\theta)$ and yaw $(\psi)$:

$$J_1(\eta_2) = \begin{bmatrix} c\psi c\theta & -s\psi c\phi + c\psi s\theta s\phi & s\psi s\phi + c\psi c\phi s\theta \\ s\psi c\theta & c\psi c\phi + s\psi s\theta s\phi & -c\psi s\phi + s\theta s\psi c\phi \\ -s\theta & c\theta s\phi & c\theta c\phi \end{bmatrix} \tag{2.5}$$

where $s(\cdot) = sin(\cdot)$ and $c(\cdot) = cos(\cdot)$.

The relation between the body-fixed angular velocity $\nu_2 = \begin{bmatrix} p & q & r \end{bmatrix}^T$ and the rate of change of the Euler angles $\begin{bmatrix} \phi & \theta & \psi \end{bmatrix}^T$ is given by:

$$\dot{\eta}_2 = J_2(\eta_2)\nu_2 \tag{2.6}$$

where $J_2(\eta_2)$ is also a transformation related to Euler angles and expressed by:

$$J_1(\eta_2) = \begin{bmatrix} 1 & s\psi t\theta & c\phi t\theta \\ 0 & c\phi & -s\phi \\ 0 & \frac{s\phi}{c\theta} & \frac{c\phi}{c\theta} \end{bmatrix} \tag{2.7}$$

where $s(\cdot) = sin(\cdot)$, $c(\cdot) = cos(\cdot)$ and $t(\cdot) = tan(\cdot)$.

It's important to notice that $J_2(\eta_2)$ is undefined for a pitch angle of $\theta = \pm 90°$since it has a singularity in that point. However this problem is not considered in the systems since the vehicles never reach this operational point.

In conclusion, the defined kinematics equation is:

$$\dot{\eta} = \begin{bmatrix} J_1(\eta_2) & 0 \\ 0 & J_2(\eta_2) \end{bmatrix} \nu \tag{2.8}$$

#### 2.1.1.3 State space representation of the AUV

A state space representation is defined to provide a compact way to model and analyze the kinematics of an AUV. The vector notation includes the position vectors $\chi$ (2.1) and $\eta$( 2.2) for the B and W-frame. The velocity vectors $\nu$ and $\dot{\eta}$ for the B and W-frame, respectively

$$\nu = \begin{bmatrix} \dot{x}_B & \dot{y}_B & \dot{z}_B & \dot{\phi}_B & \dot{\theta}_B & \dot{\psi}_B \end{bmatrix}^T = \begin{bmatrix} u & v & w & p & q & r \end{bmatrix}^T \tag{2.9}$$

and

$$\dot{\eta} = \begin{bmatrix} \dot{x} & \dot{y} & \dot{z} & \dot{\phi} & \dot{\theta} & \dot{\psi} \end{bmatrix} \tag{2.10}$$

and the force/torque vector $\varsigma$ of the thruster input,

$$\varsigma = \begin{bmatrix} \varsigma_u & \varsigma_v & \varsigma_w & \varsigma_\varphi & \varsigma_\theta & \varsigma_\psi \end{bmatrix} \tag{2.11}$$

### 2.1.2 Rigid-body dynamics

In this section, the rigid-body equations of motion in six degrees of freedom will be deduced. Newtonian and Lagragian mechanics will not be discussed. An introduction to translational and rotational equations can be found in Fossen [8] and Healey [9]. A summarized form of the kinematic and dynamic equations of motion are summarized in 2.7 which gives the relative velocity form, including current drift effects. All hydrodynamics forces and moments are expressed in relative velocity and acceleration.

$$\textbf{\textit{Surge Equation of Motion}}$$
$$m[\dot{u}_r - \nu_r r + wq - x_g(q^2 + r^2) + y_G(pq - \dot{r}) + z_G(pr + \dot{q})] + (W - B)sin\theta = X_f + m(rv_c - qw_c)$$
$$\textbf{\textit{Sway Equation of Motion}}$$
$$m[\dot{v}_r + u_r - w_r p + x_g(pq + \dot{r}) - y_G(p^2 + r^2) + z_G(qr - \dot{p})] - (W - B)cos\theta sin\varphi = Y_f + m(pw_c - ru_c)$$
$$\textbf{\textit{Heave Equation of Motion}}$$
$$m[\dot{w}_r - u_r q + v_r p + x_G(pr - \dot{q}) + y_G(qr + \dot{p}) - z_G(p^2 + q^2)] - (W - B)cos\theta cos\varphi = Z_f + m(qu_c - pv_c)$$
$$\textbf{\textit{Roll Equation of Motion}}$$
$$I_x\dot{p} + (I_z - I_y)qr + I_{xy}(pr - \dot{q}) - I_{yz}(q^2 - r^2) - I_{xz}(pq + \dot{r}) + m[y_G(\dot{w} - u_r q + v_r p) - z_G(\dot{r} + u_r r - w_r p)]$$
$$-(y_g W - y_B B)cos\theta cos\varphi + (z_G W - z_B B)cos\theta sin\varphi = K_f + my_G(u_c q - v_c p) - mz_g(w_c p - u_c r)$$
$$\textbf{\textit{Pitch Equation of Motion}}$$
$$I_y\dot{q} + (I_x - I_z)pr - I_{xy}(qr + \dot{p}) + I_{yz}(pq - \dot{r}) + I_{xz}(p^2 - r^2) - m[x_G(\dot{w}_r - u_r q + v_r p) - z_G(\dot{u}_r - vr_r + w_r q)]$$
$$+(x_G W - x_B B)cos\theta cos\varphi + (z_G W - z_B B)sin\theta = M_f + mz_G(v_c r - w_c q) - mx_G(u_c q - v_c p)$$
$$\textbf{\textit{Yaw Equation of Motion}}$$
$$I_z\dot{r} + (I_y - I_x)pq - I_{xy}(p^2 - q^2) - I_{yz}(pr + \dot{q}) + I_{xz}(qr - \dot{p}) + m[x_G(\dot{v}_r + u_r r - w_r p) - y_G(\dot{u}_r + w_r q)]$$
$$-(x_G W - x_B B)cos\theta sin\varphi - (y_G W - y_B B)sin\theta = N_f + mx_G(w_c p - u_c r) - my_G(v_c r - w_c q) \tag{2.12}$$

where $m$ is the vehicle mass, $I_x, I_y$ and $I_z$ are the moments of inertia about $x-, y-$ and $z-$ axis of the body-frame and $I_{xy}, I_{xz}$ and $I_{yz}$ are the products of inertia. $(x_G, y_G, z_G)$ are the coordinates of the vehicle's center of gravity (CG) and $(x_B, y_B, z_B)$ are the coordinates of the vehicle's center of buoyancy (CB), both expressed in the vehicle body-fixed reference frame.

#### 2.1.2.1 Hydrodynamic forces and moments

1. Radiation-Induced Forces - forces on the body when it is forced to oscillate with the wave excitation frequency and there are no incident waves. They can be identified has:

   - *added mass* due to the inertia of surrounding fluid;
   - *damping*, caused by surface friction (laminar and turbulent) and vortex shredding;
   - *restoring forces* due to Archimedes (weight and buoyancy).

2. Froude-Kriloff and Diffraction Forces - forces on the body when it is restrained from oscillating and there are incident regular waves.

The 6-DOF rigid-body dynamic equations of motion in 2.12 can therefore be expanded with the equation for the forces and moments actuating in the vehicle:

$$M\dot{\upsilon} + C(\upsilon)\upsilon + D(\upsilon)\upsilon + g(\eta) = \tau \tag{2.13}$$

$$\dot{\eta} = J(\eta_2)\upsilon \tag{2.14}$$

where M is the constant of inertia and added mass matrix, $C(\upsilon)$ is the Coriolis matrix, $D(\upsilon)$ is the damping matrix, $g(\eta)$ is the vector of restoring forces and moments and $\tau$ is the vector of the body.fixed forces from the actuators. The following sections will explain each one of the equation parameters.

The *Constant Inertia and Added Mass Matrix M* is expressed in the following way:

$$M = M_{RB} + M_A \tag{2.15}$$

where $M_{RB}$ is the rigid-body inertia matrix and $M_A$ is the added mass matrix. The added mass matrix $M_A$ represents the inertial reaction to the fluid particles surrounding the body that are accelerated by the movement of the body itself. For a rigid body in a ideal fluid, the added mass matrix will be symmetrical, i.e., $M_A = M_A^T$. Under this assumption the inertia matrix, M , will also be symmetrical, thus positive definite:

$$M = M^T > 0 \tag{2.16}$$

Hence, $M$ takes the form:

$$M = \begin{bmatrix} m - X_{\dot{u}} & -X_{\dot{v}} & -X_{\dot{w}} & -X_{\dot{p}} & mz_G - X_{\dot{q}} & -my_G - X_{\dot{r}} \\ -X_{\dot{v}} & m - Y_{\dot{v}} & -Y_{\dot{w}} & -mz_G - Y_{\dot{p}} & -Y_{\dot{q}} & mx_G - Y_{\dot{r}} \\ -X_{\dot{w}} & -Y_{\dot{w}} & m - Z_{\dot{w}} & my_G - Z_{\dot{p}} & -mx_G - Z_{\dot{q}} & -Z_{\dot{r}} \\ -X_{\dot{p}} & -mz_G - Y_{\dot{p}} & my_G - Z_{\dot{p}} & I_x - K_{\dot{p}} & -I_{xy} - K_{\dot{q}} & -I_{zx} - K_{\dot{r}} \\ mz_G - X_{\dot{q}} & -Y_{\dot{q}} & -mx_G - Z_{\dot{q}} & -I_{xy} - K_{\dot{q}} & I_y - M_{\dot{q}} & -I_{yz} - M_{\dot{r}} \\ -my_G - X_{\dot{r}} & mx_G - Y_{\dot{r}} & -Z_{\dot{r}} & -I_{zx} - K_{\dot{r}} & -I_{yz} - M_{\dot{r}} & I_z - N_{\dot{r}} \end{bmatrix} \tag{2.17}$$

This notation, based on SNAME [10], indicates the degrees of freedom on which the hydrodynamics added mass force actuates, as well the origin of it.

The *Coriolis and Centripetal Matrix* $C(v)$ is represented by:

$$C(v) = C_{RB} + C_A(v) \tag{2.18}$$

where the $C_{RB}$ consists in the rigid-body Coriolis and centripetal matrix and $C_A$ is the hydrodynamic Coriolis and centripetal matrix.

$$C_A(v) = \begin{bmatrix} 0 & 0 & 0 & 0 & -a_3 & a_2 \\ 0 & 0 & 0 & 0 & a_3 & -a_1 \\ 0 & 0 & 0 & -a_2 & a_1 & 0 \\ 0 & -a_3 & a_2 & 0 & -b_3 & b_2 \\ a_3 & 0 & -a_1 & b_3 & 0 & -b_1 \\ -a_2 & a_1 & 0 & -b_2 & b_1 & 0 \end{bmatrix} \tag{2.19}$$

where

$$\begin{aligned} a_1 &= X_{\dot{u}}u + X_{\dot{v}}v + X_{\dot{w}}w + X_{\dot{p}}p + X_{\dot{q}}q + X_{\dot{r}}r \\ a_2 &= X_{\dot{v}}u + Y_{\dot{v}}v + Y_{\dot{w}}w + Y_{\dot{p}}p + Y_{\dot{q}}q + Y_{\dot{r}}r \\ a_3 &= X_{\dot{w}}u + Y_{\dot{w}}v + Z_{\dot{w}}w + Z_{\dot{p}}p + Z_{\dot{q}}q + Z_{\dot{r}}r \\ b_1 &= X_{\dot{p}}u + Y_{\dot{p}}v + Z_{\dot{p}}w + K_{\dot{p}}p + K_{\dot{q}}q + K_{\dot{r}}r \\ b_2 &= X_{\dot{q}}u + Y_{\dot{q}}v + Z_{\dot{q}}w + K_{\dot{q}}p + M_{\dot{q}}q + M_{\dot{r}}r \\ b_3 &= X_{\dot{r}}u + Y_{\dot{r}}v + Z_{\dot{r}}w + K_{\dot{r}}p + M_{\dot{r}}q + N_{\dot{r}}r \end{aligned} \tag{2.20}$$

*Damping Matrix* $D(v)$ represents the hydrodynamic damping for ocean vehicles which is manly caused by the sum of the following components: $D_p(v)$ is the radiation-induced potential damping due to forced body oscillations, $D_s(v)$ is the linear skin friction due to laminar boundary layers and quadratic skin friction due to turbulent boundary layers. $D_w(v)$ is the wave drift damping and $D_M(v)$ is the damping due to vortex shedding, based on Morison's equation.

The *Vector of Restoring Forces and Moments* $g(\eta)$ is a representation in hydrodynamics terminology of gravitational and buoyant forces. The gravitational force $f_G$ will act through the center of gravity (CG) of the body of the vehicle and, similarly, $f_B$, the buoyant force, will act through the center of buoyancy (CB).

$$g(\eta) = \begin{bmatrix} (W-B)s\theta \\ -(W-B)c\theta s\phi \\ -(W-B)c\theta c\phi \\ -(y_G W - y_B B)c\theta c\phi + (z_G W - z_B B)c\theta s\phi \\ (Z_G W - Z_B B)s\theta + (x_G W - x_B B)c\theta c\phi \\ -(x_G W - x_B B)c\theta c\phi - (y_G W - y_B B)s\theta \end{bmatrix} \tag{2.21}$$

### 2.1.3 Current-Induced Forces and Moments

This section presents that the current-induced forces and moments can be included in dynamic equations of motion. This methods are based on the assumption that equations of motion can be represented in terms of velocity of the vehicle relative to the ocean currents and expressed in the body-fixed reference frame:

$$\nu_r = \nu - \nu_c \tag{2.22}$$

where $\nu_c = [u_c, v_c, w_c, 0, 0, 0]^T$ is a vector of non-rotational body- fixed current velocities. From the 6 degrees of freedom equations of motion 2.13 it is possible to define:

$$M\dot{\upsilon}_r + C(\upsilon_r)\upsilon_r + D(\upsilon_r)\upsilon_r + g(\eta) = \tau \tag{2.23}$$

$$\dot{\eta} = J(\eta_)\upsilon = J(\eta)(\upsilon_r + \upsilon_c) \tag{2.24}$$

and recall that its possible to compute the earth-fixed current velocity vector($v_c^e$) as follows

$$v_c^e = J(\eta)v_c \tag{2.25}$$

where $v_c^e = [u_c^e, v_c^e, w_c^e, 0, 0, 0]^T$.

Next, the kinematic equations 2.8 can be modified to include a new state variable $v_r$ and a vector $v_c^E$ describing the earth-fixed current velocity:

$$\dot{\eta} = J(\eta)v_r + v_c^E \tag{2.26}$$

## 2.2 Coordinate Systems and Transformations

### 2.2.1 Coordinate Systems

In navigation of autonomous vehicles, being them maritime, terrestrial or aerial, there are a several coordinate systems that are intensively used in design and analysis. This section, which is heavily based on Unmanned Rotorcraft Systems [11], introduces the coordinate systems studied which include

1. the geodetic coordinate system,
2. the earth-centered earth-fixed (ECEF) coordinate system,
3. the local north-east-down (NED) coordinate system,
4. the vehicle-carried NED coordinate system,
5. the body coordinate system.

The relations between the coordinate systems are also introduced for application proposes and are presented in figure 2.2 for ECEF and NED and 2.1 for Body and NED .

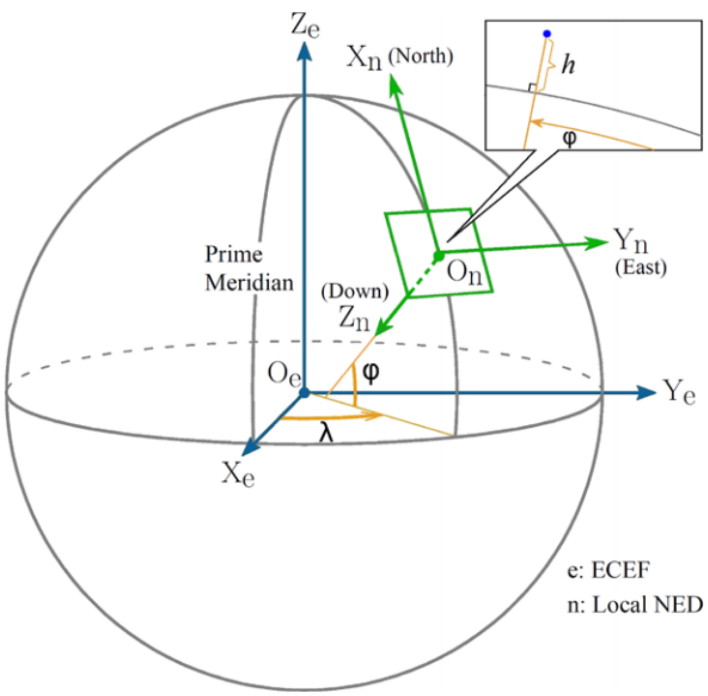


Figure 2.2: Geodetic, ECEF coordinate systems. Source: Unmanned Rotorcraft Systems

#### 2.2.1.1 Geodetic Coordinate System

The geodetic coordinate system seen in figure 2.2 is used in GPS-based applications. It characterizes a coordinate point on earth's surface with longitude($\varphi$), latitude($\lambda$) and height($h$). Longitude measures the rotational angle, from $-\pi$ to $\pi$ between the Prime Meridian and the measured point while the latitude measures the angle, from $-\frac{\pi}{2}$ to $\frac{\pi}{2}$ between the equator and the normal of the reference ellipsoid that passes through the measured point. The height (or depth) is the local vertical distance between the measured point and the reference ellipsoid. Coordinate vectors are expressed in terms of the geodetic frame are denoted by

$$P_g = \begin{pmatrix} \lambda \\ \varphi \\ h \end{pmatrix} \tag{2.27}$$

There is some important parameters that are associated with the geodetic frame that must be noted

1. the semi-major axis $R_{Ea}$,
2. the flattening factor $f$,
3. the semi-minor axis $REb$,
4. the first eccentricity $e$,

5. the meridian radius of curvature $M_E$,

6. the prime vertical radius of curvature $N_E$.

were some parameters are either defined (item 1 and 2) or derived (items 3 t o 6) and are all based on the WGS84 (world geodetic system 84, proposed in 1984 and updated in 2004 [12]) ellipsoid model. More specifically

$$R_{Ea} = 6,378,137.0m, \tag{2.28}$$

$$f = \frac{1}{298.257223563}, \tag{2.29}$$

$$R_{Eb} = R_{Ea}(1-f) = 6,356,752.0m, \tag{2.30}$$

$$e = \frac{\sqrt{R_{Ea}^2 - R_{Eb}^2}}{R_{Ea}} = 0.081181919, \tag{2.31}$$

$$M_E = \frac{R_{Ea}(1-e^2)}{(1-e^2 sin^2\varphi)^{\frac{3}{2}}}, \tag{2.32}$$

$$N_E = \frac{R_{Ea}}{\sqrt{1-e^2 sin^2\varphi}}, \tag{2.33}$$

#### 2.2.1.2 Earth-Centered Earth-Fixed Coordinate System

The ECEF coordinate system rotates with the earth, around its spin axis. As such, a fixed point on earths surface will have a fixed set of coordinates. The origin and axes of ECEF can be seen in figure 2.2 and are defined:

1. The origin $O_e$ is located at the center of the earth.

2. The Z-axis $Z_e$ is along the spin axis and points to the north pole.

3. The X-axis $X_e$ intersects the sphere of the earth at 0° latitude and 0° longitude.

4. The Y-axis $Y_e$ is orthogonal to the Z- and X-Axes based on the right hand rule.

The representation of the coordinate vectors of ECEF frame is similar to the geodetic system and is denoted by

$$P_e = \begin{pmatrix} x_e \\ y_e \\ z_e \end{pmatrix} \tag{2.34}$$

#### 2.2.1.3 Local North-East-Down Coordinate System

NED coordinate system is also known as a navigation or ground coordinate system. It's a frame fixed to earth's surface. The origin and axes, has seen in figure 2.2 are base on the WGS84 ellipsoid model and are defined has the following:

1. The origin $O_n$ is arbitrary fixed to a point on earth's surface.
2. The X-axis $X_n$ points toward the ellipsoid north (geodetic north).
3. The Y-axis $Y_n$ points toward the ellipsoid east (geodetic east).
4. The Z-axis $Z_n$ points downward along the ellipsoid normal.

The local NED frame carries a very important role in the vehicles navigation since it is normally in this frame. Coordinate vectors are $P_n$ for the position, $V_n$ for the velocity and $a_n$ for the acceleration. They are defined as:

$$P_n = \begin{pmatrix} x_n \\ y_n \\ z_n \end{pmatrix}, V_n = \begin{pmatrix} u_n \\ v_n \\ w_n \end{pmatrix}, a_e = \begin{pmatrix} a_{X,n} \\ a_{Y,n} \\ a_{Z,n} \end{pmatrix} \tag{2.35}$$

#### 2.2.1.4 Vehicle-Carried North-East-Down Coordinate System

The vehicle-carried NED system is associated with the vehicle and its origin and axes (2.1) are given by the following:

1. The origin $O_{nv}$ is located at the CG of the vehicle.
2. The X-axis $X_{nv}$ points toward the ellipsoid north (geodetic north).
3. The Y-axis $Y_{nv}$ points toward the ellipsoid east (geodetic east).
4. The Z-axis $Z_{nv}$ points downward along the ellipsoid normal.

Strictly speaking, the axis directions of the vehicle-carried NED frame vary with respect of the vehicle movement and thus are not aligned with those of the local NED frame. But since the vehicle only moves through a small region with low speed, the directional difference is completely small. As such, is reasonable to assume that the directions of the vehicle-carried and local NED coordinate systems constantly coincide with each other. The velocity and acceleration vectors of the vehicle-carried NED frame are defined

$$V_{nv} = \begin{pmatrix} u_{nv} \\ v_{nv} \\ w_{nv} \end{pmatrix}, a_{nv} = \begin{pmatrix} a_{X,nv} \\ a_{Y,nv} \\ a_{Z,nv} \end{pmatrix} \tag{2.36}$$

#### 2.2.1.5 Body Coordinate System

The body coordinate system is vehicle-carried and is directly defined on the body of the vehicle. Its origin and axes are represented in figure 2.1 and are given by the following:

1. The origin $O_b$ is located at the CG of the vehicle.
2. The X-axis $X_b$ points forward, lying in the symmetric plane of the vehicle.
3. The Y-axis $Y_b$ is starboard (right side of the vehicle).
4. The Z-axis $Z_b$ points downward to comply with the right-hand rule.

The coordinate vectors of the body frame are defined

$$V_b = \begin{pmatrix} u_b \\ v_b \\ w_b \end{pmatrix}, a_b = \begin{pmatrix} a_x \\ a_y \\ a_z \end{pmatrix} \tag{2.37}$$

### 2.2.2 Coordinate Transformations

In many applications, a coordinate transformation is needed in order to process with the navigation of the vehicle. In this section we introduce the coordinate transformations among the coordinate systems.

#### 2.2.2.1 Geodetic and ECEF Coordinate Systems

The position vector from the geodetic system to the ECEF is a step to convert GPS position measurement to the local NED system. Given a point in the geodetic system

$$P_g = \begin{pmatrix} \lambda \\ \varphi \\ h \end{pmatrix}$$

its coordinate in ECEF frame is

$$P_e = \begin{pmatrix} x_e \\ y_e \\ z_e \end{pmatrix} = \begin{pmatrix} (N_e + h)cos\varphi cos\lambda \\ (N_e + h)cos\varphi sin\lambda \\ [N_e(1-e^2)+h]sin\varphi \end{pmatrix} \tag{2.38}$$

where $e$ and $N_e$ are given in 2.30 and 2.32 respectively.

#### 2.2.2.2 ECEF and Local NED Coordinate Systems

The position transformation from ECEF to local NED is required together with the transformation from geodetic system to ECEF in order to have a complete position conversion from geodetic to local NED.

We have

$$P_N = R_{\frac{n}{e}}(P_e - P_{e,ref}) \tag{2.39}$$

where $P_{e,ref}$ is the position of the origin of NED in the ECEF coordinate system and $R_{\frac{n}{e}}$ is the rotation matrix from the ECEF frame to the local NED, which is

$$R_{\frac{n}{e}} = \begin{bmatrix} -sin\lambda_{ref}cos\lambda_{ref} & -sin\varphi_{ref}sin\lambda_{ref} & cos\varphi_{ref} \\ -sin\varphi_{ref} & cos\lambda ref & 0 \\ -cos\varphi_{ref}cos\lambda_{ref} & -cos\varphi_{ref}sin\lambda_{ref} & -sin\varphi_{ref} \end{bmatrix} \tag{2.40}$$

where $\lambda_{ref}$ and $\varphi_{ref}$ are the geodetic longitude and latitude corresponding to $P_{e,ref}$.

#### 2.2.2.3 Geodetic and Vehicle-Carried NED Coordinate System

The derivative of the kinematic relationship between geodetic position and vehicle-carried NED velocity is expressed as the following:

$$\dot{\lambda} = \frac{v_{nv}}{(N_E + h)cos\varphi} \tag{2.41}$$

$$\dot{\varphi} = \frac{u_{nv}}{M_E + h} \tag{2.42}$$

and

$$\dot{h} = -w_{nv} \tag{2.43}$$

The first two equations are derived based on spherical triangles and the third one can be obtained from the definitions of $h$ and $w_{nv}$.

The derivatives of the vehicle-carried NED velocities are

$$\dot{u_{nv}} = -\frac{v_{nv}^2 sin\varphi}{(N_E + h)cos\varphi} + \frac{u_{nv}w_{nv}}{M_E + h} + a_{mx,nv} \tag{2.44}$$

$$\dot{v_{nv}} = \frac{u_{nv}v_{nv}sin\varphi}{(N_E + h)cos\varphi} + \frac{v_{nv}w_{nv}}{N_E + h} + a_{my,nv} \tag{2.45}$$

and

$$\dot{w_{nv}} = -\frac{v_{nv}^2}{(N_E + h)} + \frac{u_{nv}^2}{M_E + h} + a_{mz,nv} \tag{2.46}$$

where

$$a_{mea,nv} = \begin{pmatrix} a_{mx,nv} \\ a_{my,nv} \\ a_{mz,nv} \end{pmatrix} \tag{2.47}$$

is a projection of $a_{mea,b}$, the acceleration measured on the body frame onto the vehicle-carried NED frame.

#### 2.2.2.4 Vehicle-Carried NED and Body Coordinate Systems

Kinematical relationships between vehicle-carried NED and the body frame of the vehicle are important to dynamics modeling and automatic movement control. We have

$$V_b = R_{b/nv} V_{nv}, \tag{2.48}$$

$$a_b = R_{b/nv} a_{nv} \tag{2.49}$$

and

$$a_{mea,b} = R_{b/nv} a_{mea,nv} \tag{2.50}$$

where $R_{b/nv}$ is the rotation matrix from vehicle-carried NED to the body frame and is given by

$$R_{b/nv} = \begin{bmatrix} c_\theta c_\psi & c_\theta s_\psi & -s_\theta \\ s_\phi s_\theta c_\psi - c_\phi s_\psi & s_\phi s_\theta s_\psi + c_\phi c_\psi & s_\phi c_\theta \\ c_\phi s_\theta c_\psi + s_\phi s_\psi & c_\phi s_\theta s_\psi - s_\phi c_\psi & c_\phi c_\theta \end{bmatrix} \tag{2.51}$$

where $s_*$ and $c_*$ denote sin(*) and cos(*), respectively.

For the rotational kinematics, the angular velocity vector $\omega^b_{b/nv}$ which describes the rotation of the vehicle-carried NED frame with respect to the body, has the following definition of Euler Angles(see 2.1.1.2). It can be expressed as

$$\omega^b_{b/nv} = \begin{pmatrix} p \\ q \\ r \end{pmatrix} = \begin{pmatrix} \dot\phi \\ 0 \\ 0 \end{pmatrix} + R_{b/int2}\left[\begin{pmatrix} 0 \\ \dot\theta \\ 0 \end{pmatrix} + R_{int2/int1}\begin{pmatrix} 0 \\ 0 \\ \dot\psi \end{pmatrix}\right] = S\begin{pmatrix} \dot\phi \\ \dot\theta \\ \dot\psi \end{pmatrix} \tag{2.52}$$

where $p$, $q$ and $r$ are the standard symbols adopted for the components $\omega^b_{b/nv}$, $R_{int2/int1}$ and $R_{b/int2}$ are given by

$$R_{int2/int1} = \begin{bmatrix} cos\theta & 0 & -sin\theta \\ 0 & 1 & 0 \\ sin\theta & 0 & cos\theta \end{bmatrix} \tag{2.53}$$

and

$$R_{b/int2} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos\phi & sin\phi \\ 0 & -sin\phi & cos\phi \end{bmatrix} \tag{2.54}$$

and lastly

$$S = \begin{bmatrix} 1 & 0 & -sin\theta \\ 0 & cos\phi & sin\phi cos\theta \\ 0 & -sin\phi & cos\phi cos\theta \end{bmatrix} \tag{2.55}$$

#### 2.2.2.5 Local and Vehicle-Carried NED Coordinate Frames

Assuming that there is no directional difference between the local and vehicle-carried NED, as mentioned at 2.2.1.4, we have

$$V_n = V_{nv}, \omega^b_{b/n} = \omega^b_{b/nv}, a_n = a_{nv}, a_{mea,n} = a_{mea,nv} \tag{2.56}$$

where $a_{mea,n}$ is the proper acceleration measured on the body frame onto the local NED frame.

## 2.3 Hybrid Systems

Nowadays, more and more real-life processes, from elevators to aircraft, are controlled by programs. Being them *reactive* programs, they must react to the environment and apply changes in real time. A *hybrid* system represents a discrete program within an analog environment. Hybrid Automata are generalized finite-state machines for modeling hybrid systems. Discrete transitions of program are modeled by a change in the program counter, which ranges over a finite set of control locations. Hybrid systems have been extensively use and because of that, its possible to have a more in-depth study where Varaiya *et. al.* have an extended published work about Hybrid Systems ([13],[14],[15]) and Hespanha [16] class notes. The Hybrid automaton model presented next [17] is a generalization of other models introduced in literature(Alur *et. al.* [18] and Puri *et. al.* [19]).

A hybrid transition system is a tuple

$$H = (Q, R^n, \sum, E, \Phi) \tag{2.57}$$

where Q is the finite set of discrete states and $R^n$ is the set of continuous state. $\sum$ is the finite set of discrete events.

$$E \subset Q \times P(R^n) \times \sum \times \{R^n \rightarrow R^n\} \times Q \tag{2.58}$$

E is the finite set of edges[1]. The edges model of the discrete event dynamics of the system. An edge $e \in E$ is denoted as

$$(q_e, X_e, V_e, r_e, q_e^{new}) \tag{2.59}$$

and is enabled when the discrete state is $q_e$ and the continuous state is in $X_e$. When a transition through $e$ is taken, the event $V_e \in \sum$ is accepted by the system. The continuous state is then reset according to the map $r_e$, and the system enters the discrete state $q_e^{new}$.

$$\Phi = \{F_q : \mathbf{R}^n \to P(\mathbf{R})^n \backslash 0 | q \in Q\} \tag{2.60}$$

is a set of differential inclusions that the model the continuous dynamics of the system. When the discrete state is $q$, the continuous state evolves according to the differential inclusion.

A hybrid automaton essentially consists of a graph with discrete states as vertices's and with edges between the discrete states.

Let's take figure 2.3 has an example.

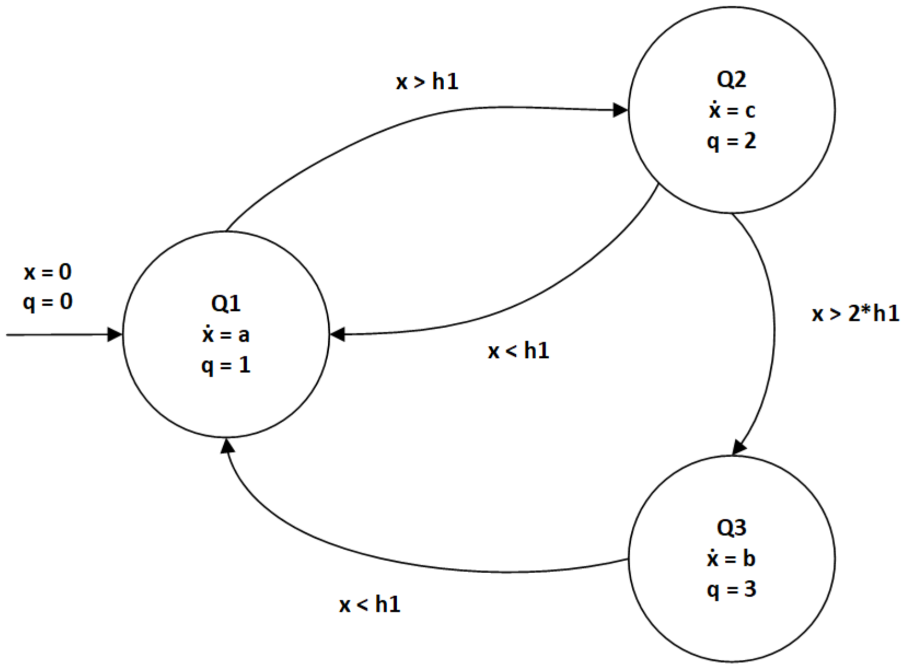


Figure 2.3: Example of a Hybrid Automaton

Each discrete state (Q) is labeled with a specific differential inclusion ($\Phi$) and every edge($E$) is labeled with a guard condition and jump relation. The state is pair $(q,x)$ where $q$ is the discrete state and $x \in \mathbf{R}^n$ is the continuous state. The initial state is $(q_0, x_0)$ and the trajectory evolves with the discrete state remaining constant and the continuous state $x$ evolving accordingly with the differential inclusion expressed in the discrete state. When the guard condition of an edge from the discrete state $q_i$ to other discrete state $q_k$ is satisfied a jump can be performed to a new state $q_k$.

[1] P($\cdot$) denotes the power set(or, the set of all subsets) of ($\cdot$)

When performing the jump, the continuous state may get initialized to a new value $x_k$ being this the reset condition. The new state pair is $(q_k, x_k)$. The continuous state $x$ now moves with the new differential inclusion. This is the general state transition algorithm for hybrid systems.

Considering again figure 2.3, now has an example, the state transition will be:

1. The hybrid system starts with a continuous state $x = 0$ and a discrete state $q = 0$ and because of that it enters the **Q1** state.

2. Entering the state **Q1** x will be $\dot{x} = a$ and the discrete state $q$ will be $q = 1$ and will have a guard condition between states **Q1** and **Q2** of $x > h_1$.

3. If the state of the system increases in a way that the limit is overflown, then the state will jump to **Q2** having $\dot{x} = c, q = 2$.

4. In case of some disturbance happens in the system where $x > 2h_1$ then the state will jump to **Q3** updating $\dot{x} = b$ and $q = 3$.

5. When the state of the system returns to normal levels ($x < h_1$) the discrete state returns to **Q1** either from **Q2** or **Q3**.

## 2.4 LSTS Toolchain

There have been a huge development on sub-aquatic autonomous systems that need to be easily controlled remotely. The Underwater Systems and Technologies Laboratory (LSTS) has developed different algorithms and technologies that allow to operate this vehicles in a simple and efficient way to the operator. The toolchain of LSTS consists in a control architecture which includes:

- Neptus.
- Inter-Module Communication (IMC).
- DUNE Unified Navigation Environment (DUNE).

The figure 2.4 show the common scheme of actuation of all parts of the toolchain.

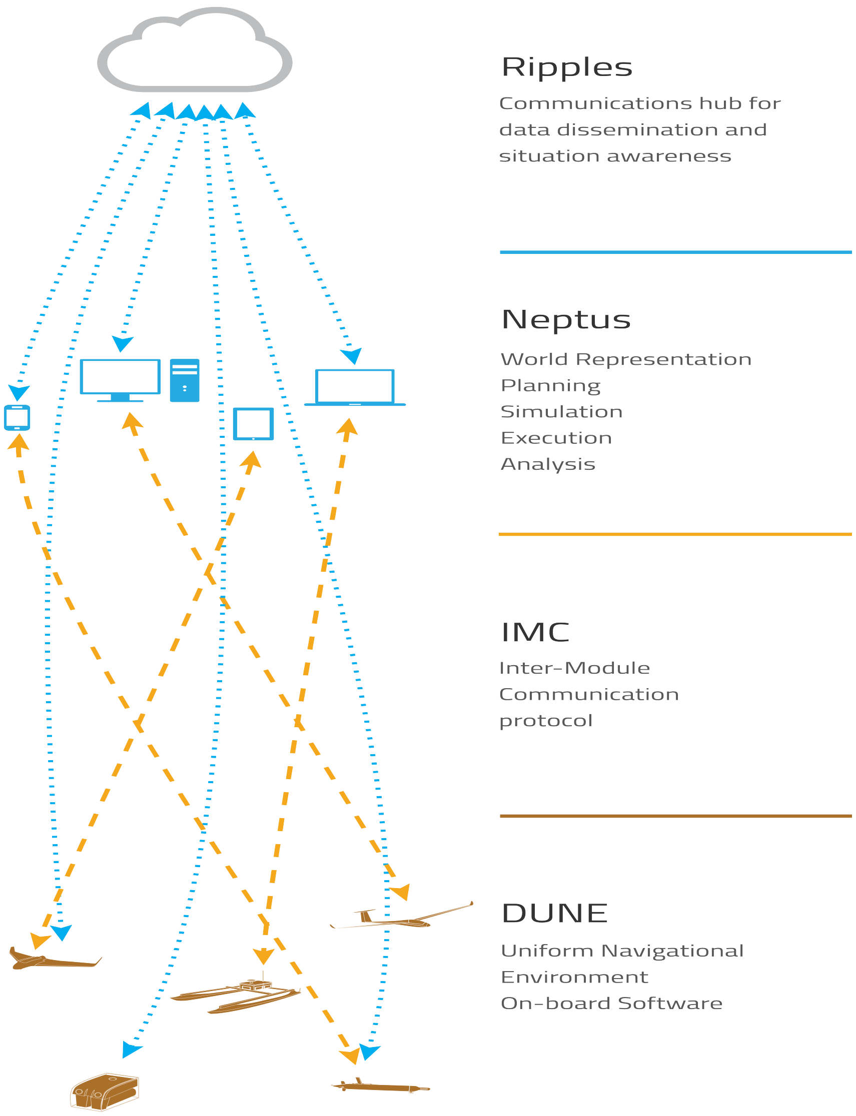


Figure 2.4: Structured top-down view of the LSTS Toolchain. Source: LSTS

### 2.4.1 Neptus

Neptus is a *Command and Control* software used to control and monitor the autonomous systems. It provides basic functions by using plugins where there is possible to have a communication infrastructure, several layout mechanisms, means of showing notifications to the operator of the mission and a map that can be extended with new layers and interaction mechanisms. This results in consoles supporting the full extent of mission life-cycles in an integrated interface. It is written in Java and it currently runs in Linux and Microsoft Windows operating systems. A typical mission life-cycle executed in Neptus comprises three phases:

1. Planning

2. Execution
3. Review

The *planning phase* generally is performed prior to the mission. The operator, equipped with the mission objectives and the knowledge of possible obstacles, depths, tides, traffic, choose the best location for the command center, communication placement and location aids and starts doing rough simulations of the mission plans.

The figure 2.5 shows and example of maneuver planning for a mission, in Neptus.

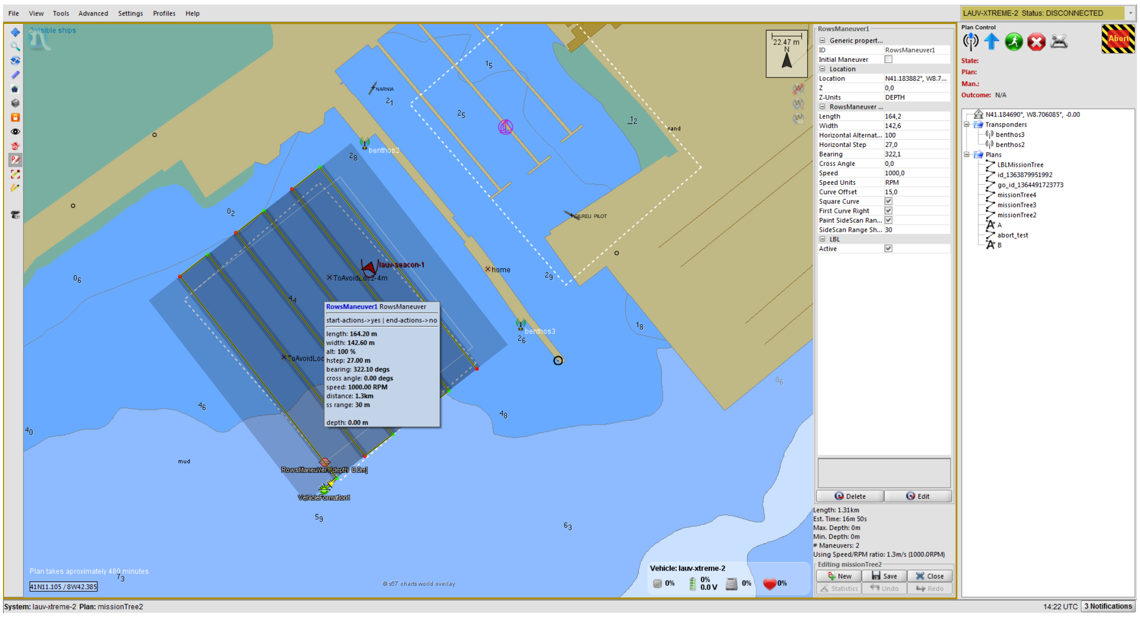


Figure 2.5: Mission planning. Source: LSTS

In *execution phase*, the operator is in charge of preparing the vehicles for deployment, monitor the systems telemetry and execute/adapt the mission plans. The figure 2.6 shows the execution and all the important parts to manage to successfully perform the mission.

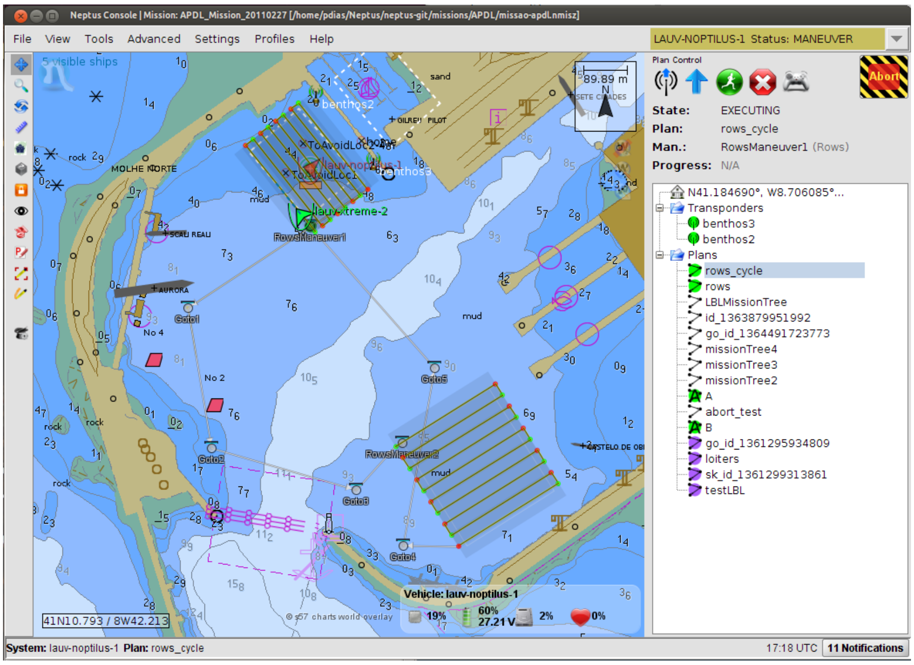


Figure 2.6: Mission execution. Source: LSTS

The *review and analysis phase* (MRA) takes place on site or after concluded the mission. It serves to process and analyzed the collected data in order to compile mission results or evaluate individual plan execution to adjust and re-plan to achieve another desired outcome of the mission. In this phase, logs are generated with the information required by the operator, for example, *sidescan* imagery has seen on figure 2.7 and vehicles path, on figure 2.8.

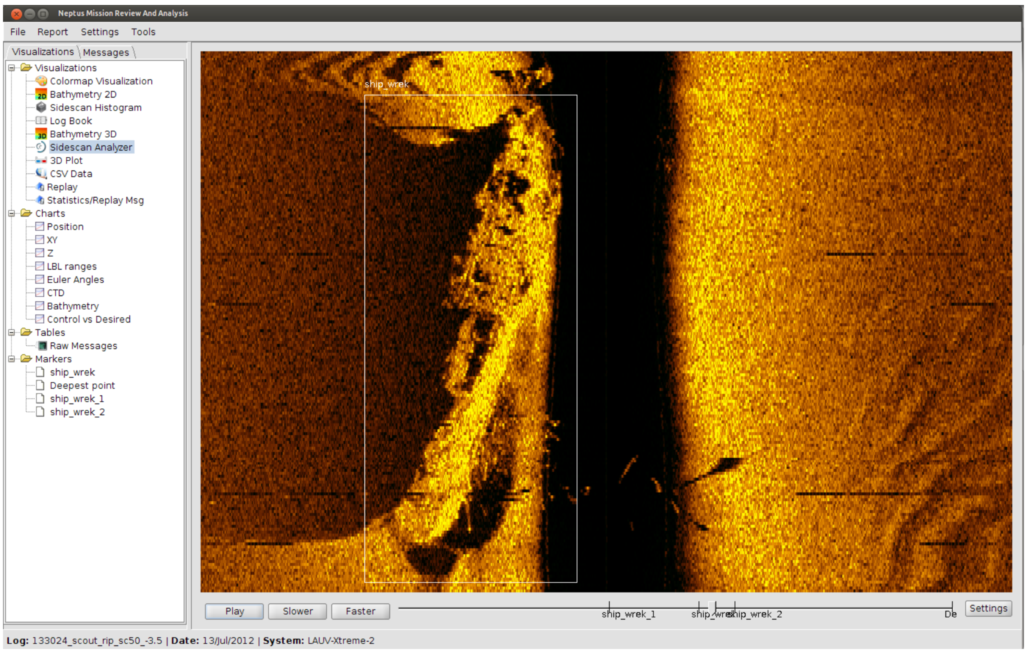


Figure 2.7: Sidescan imagery analysis. Source: LSTS

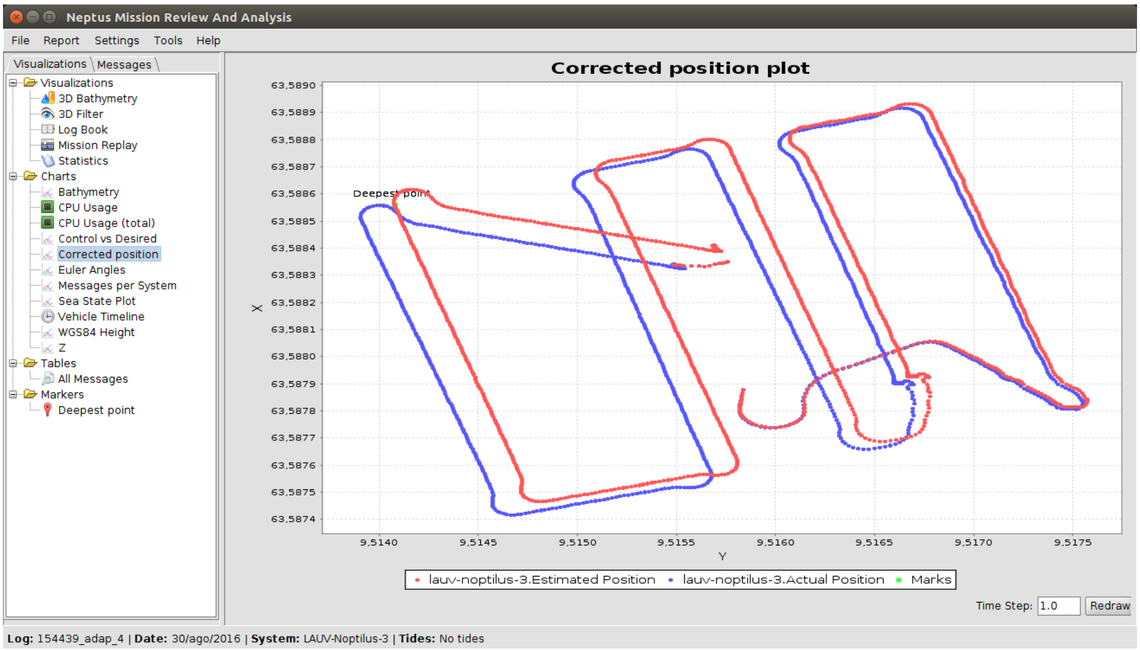


Figure 2.8: Vehicle estimated and real position. Source: LSTS

### 2.4.2 Inter-Module Communication

*Inter-Module Communication* (IMC) protocol is a message-oriented protocol designed and implemented to build interconnected systems of vehicles, sensors and human operators. Has they pursue a common goal of cooperatively exchange real-time information about the environment or update objectives, IMC abstracts the hardware and communication by providing a shared set of messages that can be serialized and transferred over different means. Native support can be automatically generated for different programming languages resulting in an optimized code that can be used both network nodes and inter-process or inter-thread communication. An example of communication between systems can be seen in figure 2.9.

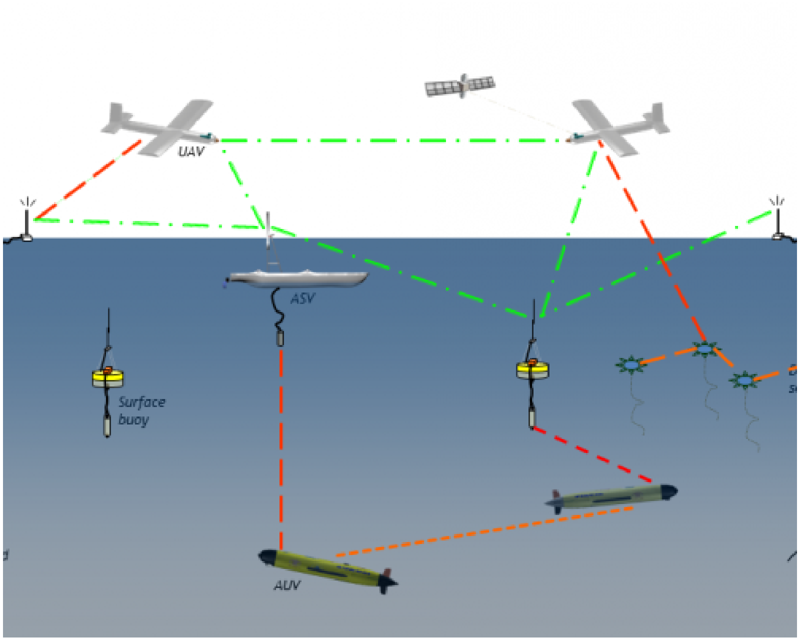


Figure 2.9: IMC communications between multiple systems. Source: LSTS

### 2.4.3 DUNE Unified Navigation Environment

*DUNE Unified Navigation Environment*(DUNE) is the on-board running on the vehicle, which is responsible to every interaction with sensors, payload, actuators, communications, navigation, control, maneuvering, plan execution and vehicle supervision. It is CPU architecture and operating system independent and due to its modularity and versatility, DUNE does not only run in ASVs, ROVs,AUVs, and UAVs but also in LSTS *Manta* communications gateways. DUNE's architecture is a collection of tasks, hierarchically structured that usually run in separate threads of execution. They can communicate with each other using the IMC protocol by forwarding IMC messages from the producers to the consumers. Each task follows a common life-cycle and also has method handlers for all messages it consumes. The figure 2.10 shows a base example of DUNE architecture.

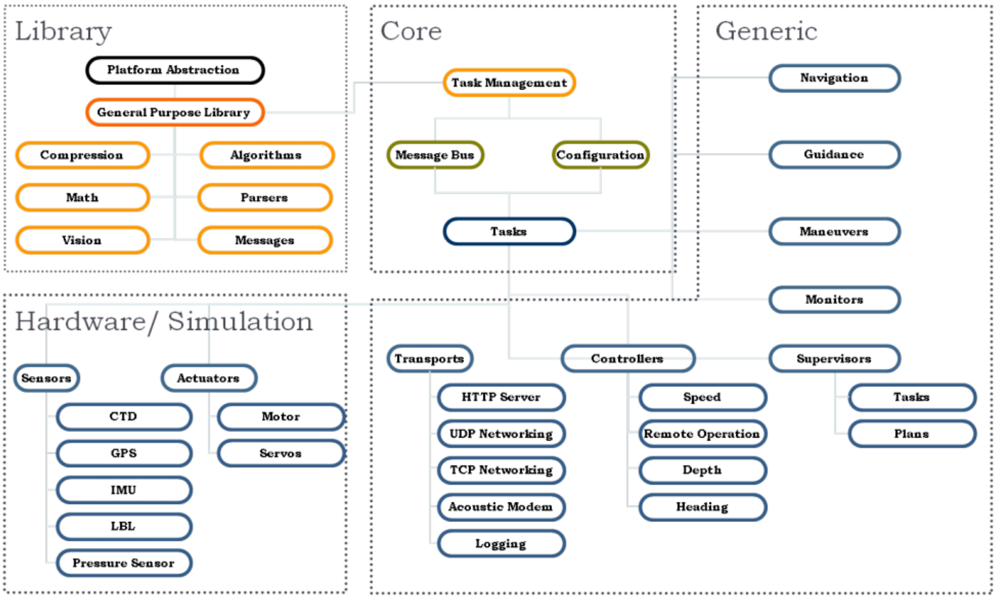


Figure 2.10: DUNE architecture. Source: LSTS

## 2.5 Conclusions

In this chapter we discussed an overview of models of marine vehicles (2.1). Several system of equations have been addressed in the kinematics section (2.1.1) where was described the coordinate frames associated with the movement of the vehicle and it's position, Euler angles, where were presented the equations associated with the body-fixed reference frame to the earth-fixed reference frame. A deduction was made towards the rigid-body equation of motion in (2.1.2). It was presented Current-Induced Forces and Moments in (2.1.3) where the equations of motion were represented in terms of velocity of the vehicle relative to the ocean currents.

It was presented a simple explanation about coordinate systems of autonomous vehicles (2.2) where we show the existing coordinate systems (2.2.1) and the possible transformations between them (2.2.2).

A brief explanation about Hybrid Systems (2.3) was shown in order to present how it is possible to describe a real-life process through a simple modeled program.

Finally, the LSTS toolchain was described in (2.4) where was made a brief analysis of the various components of the toolchain, showing the operation of the interface Neptus 2.4.1, how IMC protocol (2.4.2) works towards a multi-system communication and how DUNE (2.4.3) manages to make a bridge between low level components like actuators and sensors, parse the information through the various tasks and send it to a higher level tasks.

# Chapter 3

# State of the Art in Autonomous Docking Systems

## 3.1 Introduction

There is already a great number of underwater docking systems developed by the research community, although some are still at present at prototype stage of development. Improvements are being developed to have a better performance of energy storage technology, precision of navigation instruments and reliability of acoustic communications. This enables progress on the development of homing/docking systems throughout the last two decades. Such improvements, made a profound impact on the ocean exploration.

This overview address the various techniques implements towards the development of docking systems.

One of the main problems for extended oceanic exploration is the shortcoming of the present technology, e.g. unavailability of affordable as suitable platforms that can be deployed for various science missions. The Autonomous Docking Systems (ADS) enhances the capability of underwater exploration and data collection by increasing the level of autonomy of the AUVs. With this, the AUVs can be deployed for extended periods of time, data can be easily transferred and batteries can be recharged.

## 3.2 Overview of underwater docking systems

In literature, some of the works specifically deal with docking maneuvers where some take homing maneuvers into consideration. With this, the following definitions are given the intention to clarify the difference between docking and homing in ADS systems.

**Docking** - As the name suggests, underwater docking is the final sequence of the maneuver of the underwater vehicle to dock on the docking station, being the former stationary or in motion. Such maneuver, typically is initiated between 50 to 500 meters from the dock.

**Homing** - This is the last but one sequence of the maneuver to perform the docking. This phase is primarily used to guide the vehicle to an optimal position, within the range of the sensors placed on the dock. Such phase, usually is initiated from 500m to 2Kms from the dock. Once the communication between the dock and the vehicle is established, the docking maneuver is initiated.

### 3.2.1 Existing docking systems and approaches

**"A Docking System for REMUS, an Autonomous Underwater Vehicle" - Stokey, Purcell, and *et. al*.**

Autonomous docking using Ultra-Short Baseline (USBL)[1] acoustic homing array were demonstrated by Stokey, Purcell, and *et. al* [3] when they developed a docking system to be used on REMUS (Remote Environmental Monitoring Unit), low cost AUV designed by the Oceanographic Systems Laboratory at Woods Hole Oceanographic Institution (WHOI). Their work discusses solutions for enabling the vehicle to dock inside a stationary conical shaped docking station, as seen in figure 3.1, where acoustically they find and home the AUV to the docking station. They managed to build procedures for mechanically latching the vehicle to the dock, applied some electro-mechanical techniques for power and data transfer from the docking system to the vehicle, and download data remotely and upload new mission without the need to open the vehicle casing.

They also developed and algorithm for the docking sequence using state machines. This algorithm uses the technique of way-point following and minimization of cross-track error[2]. The docking maneuver happens in the following sequence: First the vehicle navigates to a position fifty meters from the dock along the track into dock. Once the criteria has been met, the vehicle attempts to follow the path leading into the dock. When the vehicle determines that is entering the dock it straightens the fins out, and continues thrusting at constant RPM for fifteen seconds. This period forces the vehicle all the way into the guide tube. This approach is important because it addresses all the components of a typical docking system, including the dock, charging mechanism and communication circuitry, AUV navigation and the vehicle software and docking algorithm. Some disadvantages with the system are a navigational problem, since the vehicle could not merely head towards the dock. The vehicle had to orient itself on the proper glide path. The system was tested at Woods Hole harbor. Some disadvantages with the system are a navigational problem, since the vehicle could not merely head towards the dock. The vehicle had to orient itself on the proper glide path. This docking system did not consider a model of the AUV nor the ocean currents in the control hoop, therefore it was not a robust control method. Thus, convergence was not guaranteed.

The work is further explained in [20] and [21].

[1]USBL is a method of underwater positioning. A complete USBL system, consists in a transceiver, which is typically 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.

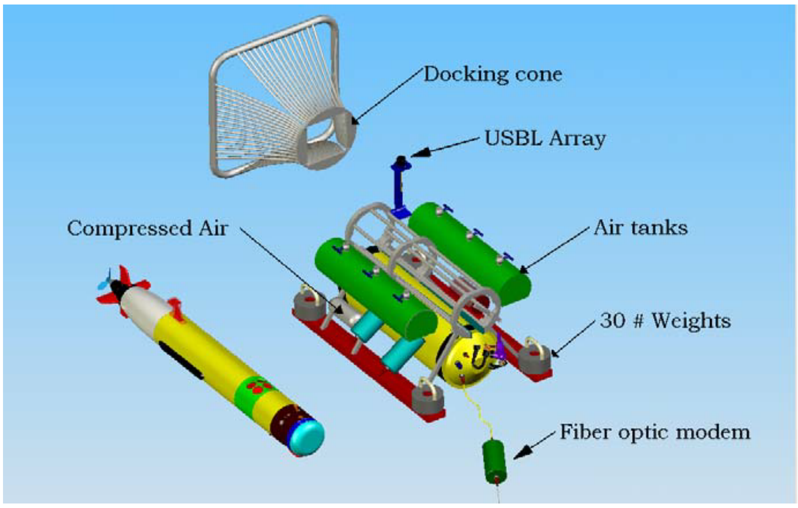


Figure 3.1: Docking system for Remus AUV. Source: Stokey *et. al*.

**"Autonomous Underwater Vehicle Homing/Docking via Electromagnetic Guidance" - Feezor *et. al.***

An electromagnetic (EM) homing system was developed by Feezor *et. al.* [5] for a AUV named *SeaGrant Odyssey IIb* to dock inside a stationary conical shaped docking station. The docking station had embedded systems that can emit magnetic fields strong enough to be accurately detected by the AUV and thus, perform the docking. The system would achieve a precision up to 20cm to the dock. Still the maximum range of the Electro Magnetic (EM) system was limited to twenty-five or thirty meters. In this approach, there were no communications between the underwater vehicle and the dock.

The docking sequence follows: the AUV was programmed to travel outbound from the launch point for sixty seconds, execute one hundred and eighty degrees turn to point towards the dock station and travel back towards the ship and dock. The AUV remains in *dead reckoning*[3] until the magnetic field is sensed. A series of tests were conducted in Buzzards Bay, where they failed when the AUV was aligned more than thirty degrees off the dock the dock axis when acquired the EM signal. In this cases, the AUV slipped during the turn to the dock and hit the outside edge of the dock during homing.

[3]Dead Reckoning is the process of calculating one's current position, based on a previous position and the estimated speed over elapsed time an course.

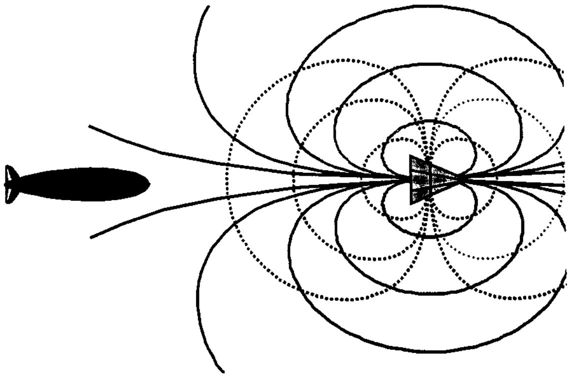

Figure 3.2: Diagram of EM homing system showing the field lines and the dock. Source: Feezor *et. al.*

**"Underwater Docking of Autonomous Undersea Vehicles using Optical Terminal Guidance" - Cowen, Steve and Briest**

Cowen *et. al.* [22] presents an optical terminal guidance system which tracks the light source provided by the dock, using two AUVs for sea trials, more specifically, the SeaGrant Odyssey IIb and the NRaD Flying Plug. In this approach, the light is tracked by the vehicles. It was demonstrated to be very accurate and robust for the vehicle terminal guidance during field operations and provided target accuracy of one centimeter under real-world conditions even with present disturbances. The control of the vehicle was achieved through the application of a conventional closed loop Proportional–Integral–Derivative (PID) control. One of the main disadvantages of this technique is that the sunlight would interfere with docking when on shallow water.

**"Docking for an autonomous ocean sampling network" - Singh, Bellingham, Hover *et.al.***

All the previous systems use a cone dock for the docking system but Singh, Bellingham, Hover *et.al.* [2] have tested an omni-directional docking system for the Odyssey vehicle (figure 3.3) using USBL system for AUV to approach the dock from any direction by determining the range and bearing of the transponder mounted on the dock. After the vehicle reached the dock, a latch mechanism would lock to a pole mounted on the dock. The advantage of system is the robustness to errors, being them the presence of ocean currents and magnetic anomalies. This work presented solutions for several failure model to ensure reliability of the system, including failure of a task to complete, communications failure, mechanical failures, conflicting sensor data, missing AUV and software lockups. They developed a layered hierarchical control architecture for 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 four different states in a higher level of abstraction:

- **State 0** - Power on sequence of the vehicle.
- **State 1** - Vehicle ready for mission, docked in the Docking station.

- **State 2** - Docking Station is empty, vehicle in mission.

- **State 3** - Vehicle ends mission, return to Docking Station.

An USBL system on the vehicle is used to calculate the azimuth[4] and elevation relative to the docking station. The Line-of-Sight (LOS) technique, which is used in the homing algorithm, works by nullifying the bearing[5] to the docking station. A PID control loop ensures that the heading $\sigma$ follows the desired heading $\psi_d$. The algorithm for the homing system is explained as follows: homing is typically initiated at a distance of one hundred to two hundred meters away from the dock. When the homing beacon is detected, the vehicle attempts to null the bearing to the dock. This approach was successfully tested with cross currents to test the robustness of the system in presence of disturbances.

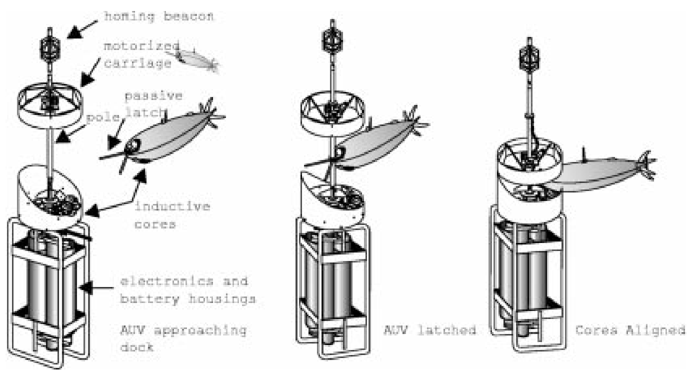


Figure 3.3: The AUV centric components of the docking system. The AUV passively latches onto a pole and a aligning inductive cores on the AUV and the dock to enable data and motorized assembly drives down power transfer. Source: Singh *et. al.*

**Autonomous docking for intervention-auvs using sonar and video-based real-time 3d pose estimation - Evans *et.al.***

This approach to a docking system, uses a sonar and video sensor processing techniques for real-time control of the AUV to perform tracking and three dimensions pose. Developed by Evans *et. al.* [6] it was developed for Intervention-AUV (I-AUV), the Autonomous Light Intervention Vehicle (ALIVE). Its purpose was to latch the vehicle onto a fixed sub-sea structure. A number of techniques implemented to guarantee a precise positioning on the docking sequence are what follows: real-time sonar-based feature tracking, video-based position control, Inertial Navigation Systems (INS) and video-base Dynamic Positioning (DP) and high-bandwidth acoustic video transmission. This comprises a three stages process:

1. **Transit** - This vehicle is monitored at the surface station by a periodic data acquisition through the acoustic modems.

[4] Horizontal angular distance between the heading of the vehicle and the reference direction.
[5] Angular direction measured from one's position to another geographical reference lines.

2. **Approach** - In this phase, the docking system uses the sonar scans to continuously track the vehicles position relative to the intervention docking panel. The vehicle gather this information and uses it to maneuver within a safe zone of approximately two a three meters from the panel. Once close enough, is trigger the video systems to start, a begins the docking phase,

3. **Docking** - When in the docking phase, sensory fusion of the sonar and video control system get the vehicle stabilized within ten centimeters from the panel. Finally, the docking manipulators extended and dock the vehicle through a latch mechanism.

The results obtain from this approach where good when done in deep water trials.

**Docking control system for a 54-cm-diameter (21-in) AUV - McEwen *et.al.***

At Monterey Bay Aquarium Research Institute (MBARI), McEwen *et. al.* [4] proposed a solution of a docking system for a fifty-four-centimeter-diameter AUV and a docking station (see figure 3.4). The AUV proceeds to the homing sequence using USBL. The docking station is a benthic[6], fixed-heading cone. The homing and docking sequence is the following:

1. **Locate and home to the docking station** - the vehicle locates the docking station and proceeds with the homing sequence using pursuit guidance (LoS Path following). This keep the heading of the vehicle control system pointing towards the beacon on the station. This method doesn't compensate for external disturbances, since ocean current can be blown downwind while making the approach. The main advantage is that is keeps the USBL pointed at the beacon for maximum strength.

2. **Compute a position fix** - While USBL keeps a good signal strength with the beacon, the vehicle uses its compass heading and USBL bearing and range to compute a position fix.

3. **Fly to the start of the final approach path** - the approach path is along the cone centerline, and begins about three hundred meters out. The disadvantage of this method is that it 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 will correct external disturbances since it acquires a drift correction angle if the ocean current has a lateral component. The vehicle slows down to one meter per second at two hundred meters from the dock. This has two purposes: first, it allows the control loop to zero the cross-track error and second, it prevents the vehicle from hitting on the dock with too much force.

5. **Latch the vehicle on the dock** - it uses a inductive position sensor and an Ethernet contact to determine if it has fully entered. Then it raises a peg and latches the vehicle.

[6] Stays on the bottom of the body of water.

The system was successfully tested in the Monterey Inner Shelf Observatory and operated by the Naval Postgraduate School.

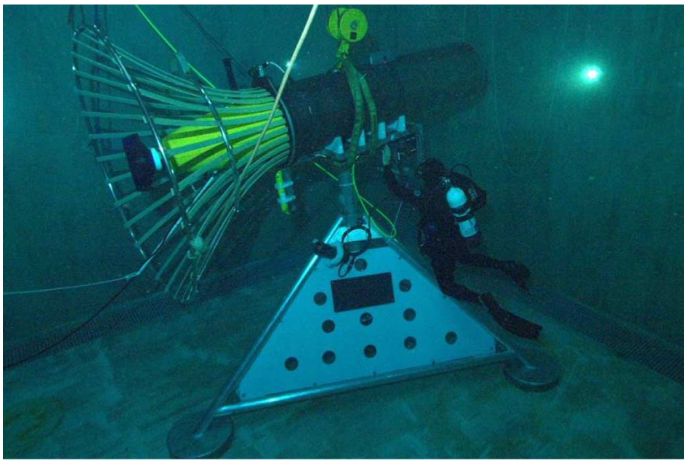

Figure 3.4: AUV in the dock in a seawater test tank. Source: McEwen *et. al.*

### A Docking and Control System for an Autonomous Underwater Vehicle - Lee *et.al.*

Lee *et.al.* [23] presented a docking system for an AUV to dock in an underwater docking station with a camera. Here, an optical flow model of a camera is mounted on the AUV, where a charge-coupled device (CCD) camera was installed on the nose of the AUV. They combined the optical flow equation of the camera with the AUV linearized equations of motion (EOM), deriving the state equation for servoing AUV. With this, they merge the AUV model with the optical flow equation, allowing to obtain more accurate control of the docking sequence. The mathematical model of the visual servoing AUV included a system of disturbances vector, for modeling errors and uncertainties of the AUV correction and white noise vector for measurement disturbances.

The control objective was to move the AUV to the docking station while having the camera centered on the target point. A control law was designed based on optimal one-step ahead predictive controller by minimizing a cost function, reflecting the distance between $y(k+1)$ and $y_d(k+1)$ where $y_d$ is the desired position in the CCD plane for an object and the attitude of the AUV in one step ahead. In the simulation they neglected the roll-induced($\theta$) velocities.

Numerical integration was conducted using the Euler method.

The results were presented by docking the AUV to a target station using the six degrees of freedom (DOF) non-linear equations of REMUS of WHOI and CCD camera.

### Experiments on vision guided docking of an autonomous underwater vehicle using one camera - Park, Jun, Lee *et. al.*

Park, Jun, Lee *et. al.* expanded the work in [23] where they present a vision guided underwater docking algorithm for an AUV, applying light in the dock to serve as a visual guide of the vehicle during the last state of docking sequence (see figure 3.5). The dock remains stationary during the docking maneuver of the AUV. This algorithm allows the vehicle to identify the dock lights, eliminating luminary noises and successfully estimate both the center of the dock and the distance to it. The control was implemented based on the decoupled equations of motion for steering and diving by *Healey* and *Lienard*[24]. To track heading and depth, a conventional PID control loop was applied. The developed algorithm was based on pure pursuit guidance law. In this, there is no way to compensate external disturbances such, e.g. ocean current. No alignment of the AUV's heading with the dock direction was applied.

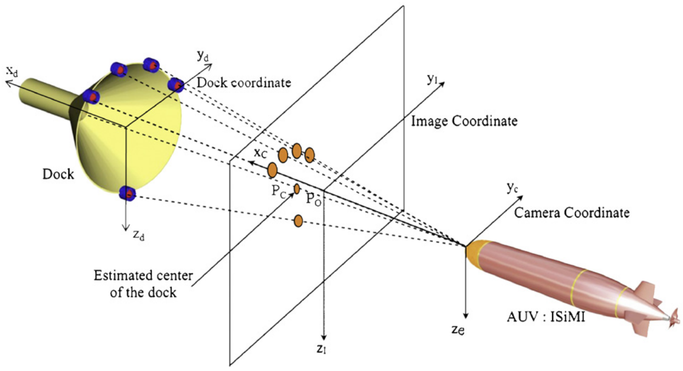


Figure 3.5: Vision guided docking system - Source: Park *et. al.*

**Improvement of vision guided underwater docking for small AUV ISiMI - Park *et.al.***

In [25], they developed a conceptual idea to overcome the problems presented before. If there is an environment disturbance, i.e ocean current, the AUV approaches to the dock with a side slope angle ($\beta$). Since the AUV is an under-actuated system, the agreement of the three state, being vehicle heading, course and dock heading, at the moment of the docking, is impossible. They assumed that a final approach and docking with the side-slip angle is more dangerous than the docking discrepancy between the course and the dock heading. It is assumed that the dock heading is fixed at $\psi_{dock} = 0$ and the AUV know his own heading. The current is also assumed to be regular and uniform. Only lateral current was considered. This was to generate a cross-track error to compensate the effects of cross currents, combined with a predetermined heading relative to the ocean currents that will drive the state of the AUV to the final state inside the docking station. The necessary equations to derive the references for the cross-track errors and desired heading ($\psi_{AUV}$) are presented. Figure 3.6 presented the compensation of the effect of lateral currents.

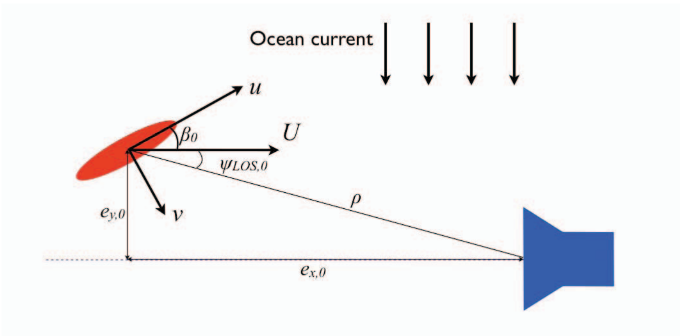


Figure 3.6: Ocean current compensation - Source: Park *et. al.*

**Control of underwater vehicles on autonomous docking maneuvers - Braga**

A control solution for docking of underwater vehicles is given by Braga [7]. That is, a hierarchical architecture of control is introduced specifically for the problem introduced. *Hybrid Automata* is used for the high level control to supervise and execute the basic maneuvers. Medium level controllers execute the way-point tracking of reference signals generated by medium level controllers. In this approach, the docking station is considered to be applied on a Remotely Operated Vehicle (ROV).

## 3.3 Acoustic positioning systems

Acoustic positioning systems exist to provide relative positioning, underwater. These systems have a varying levels of capability as a Dynamic Positioning (DP). They can provide a position reference from 3,700m with an absolute accuracy of 3-5m and a relative accuracy of <2m. Also there are available systems with shorter range, higher resolution. The table 3.1 shows the relation between the frequency bands and the maximum range of the systems.

| | Frequency Range | Maximum range | Typical relative, static accuracy |
|---|---|---|---|
| Low frequency (LF) | 8kHz to 16kHz | $\sim$ 10km | 2m to 5m |
| Medium frequency (MF) | 18kHz to 36kHz | 2km to 3.5km | 0.25m to 1m |
| High frequency (HF) | 30kHz to 60kHz | 1, 500m | 0.15m to 0.25m |
| Extra high frequency (EHF) | 50kHz to 110kHz | <1, 000m | <0.05m |
| Very high frequency (VHF) | 200kHz to 300kHz | <100m | <0.01m |

Table 3.1: Frequency Bands and Maximum Range - Source: Vickery

The decision of using a particular frequency band is made based on the application at hand. The lower the frequency range used for interrogation, the higher the effective range, and lower the accuracy. The depth of operation is central to this decision. The deeper the area of operation the lower the frequency band used. The simple reason being higher frequencies are attenuated more than lower ones.

The distance between acoustic baselines is what defines the acoustic positioning system. Vickery [26] presents the three primary types of acoustic positioning systems:

- LBL - Long Baseline (Baseline Length from 100m to 6,000m+ )
- SBL - Sort Baseline (Baseline Length from 20m to 50m )
- USBL - Ultra Short Baseline (Baseline Length <10cm )

### 3.3.1 Long Baseline

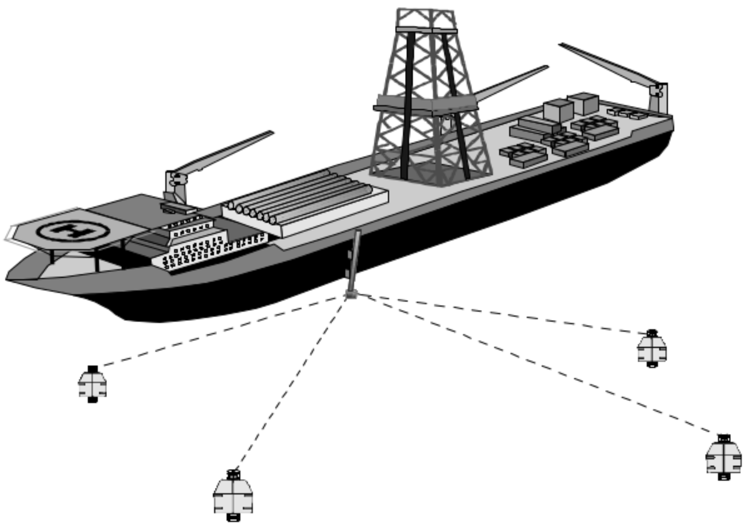

Figure 3.7: Long Baseline - Source: Vickery

As said previously, Long Baseline (LBL) system take their name from the distance between seabed beacons. The conventional LBL acoustic position systems normally use a Kalman Filter[7] correction to handle the problem of positional error has described by Cheng [27]. A typical configuration of the system, consists in one transceiver and at least three transponders. The transceiver is mounted on a AUV or surface vessel which is the target to be positioned. The transponder form an array (see figure 3.7). This transponders are deployed before, on the sea-floor and it's position are know precisely. To determine the location of the vehicle, it usually follows the sequence:

1. The vehicle emits and acoustic pulse from its transceiver.
2. The pulse travels through the water to each LBL transponder.

[7]Kalman filter, also known as Linear Quadratic Estimation is an algorithm that uses a series of measurements observed over time, with noise and produces estimates of unknown variables.

3. The transponders detect the signal and respond with a acoustic pulse of a unique frequency.

4. The transponders pulse return through the water to the vehicle's transceiver.

5. The vehicle's processor determines the travel time of each transponder and calculates the range of each with a given sound speed of water.

There are some advantages or disadvantages about the LBL system that are described by Vickery [26] which the advantages follows:

- Very good position accuracy independent of the water depth.
- Observation redundancy.
- Can provide high relative accuracy positioning over large areas.
- Small transducer - only one deployment machine/pole.

and disadvantages are:

- Complex system requiring expert operators.
- Large arrays of expensive equipment.
- Operational time consumed for deployment/recovery

### 3.3.2 Short Baseline

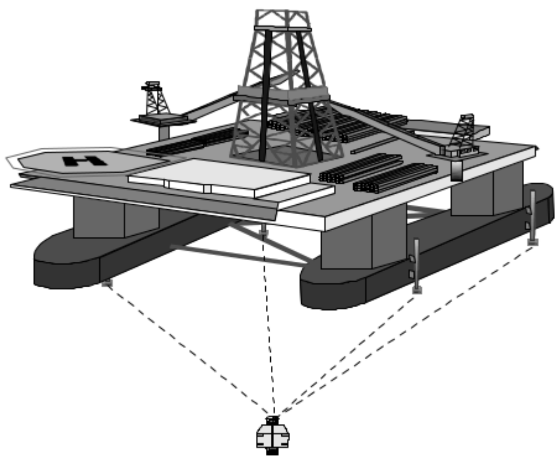

Figure 3.8: Short Baseline - Source: Vickery

Short Baseline (SBL) is similar to Long Baseline technique. It does not require any sea-floor mounted instruments since three or more transceivers are installed on the hull of a ship or on a surface platform (see figure 3.8). The transponder is attached to the AUV to be positioned. One of the transceivers sends out an acoustic signal and the transponder responds it with another signal on a different frequency. This signal is received by the transceiver array and the two-way time-of-flight from the transponder to the transceiver is measured and converted to a range at sound speed on site. The AUV's position is then known using the trilateration method[8].

The SBL positioning accuracy improves with the operating range and the spacing between the transceivers on the surface platform. Compared to LBL systems, the low system complexity makes SBL easy to use. Since the transceivers are applied in the surface platform, there is no need to deploy them on the sea-floor. The advantages of the SBL system proposed by Vickery [26] are:

- Good update rate when used with a pinger.
- Good range accuracy with time of flight system.
- Spatial redundancy built-in.
- Ship based system - no need to deploy transponders on the sea-floor.
- Low system complexity makes SBL an easy tool to use.
- Small transducers values.

and the disadvantages:

- System needs large baselines for accuracy in deep water (>30m).
- Very good dry dock/structure calibration required.
- Detailed offshore calibration of system required - usually not rigorously completed.
- Absolute position accuracy depends on additional sensors - ship's gyro and vertical reference unit.
- >3 transceiver deployment poles/machines needed.

[8] Trilateration is the process to determining absolute or relative locations of points by measurement of distances, using geometry of circles, triangles or spheres.

### 3.3.3 Ultra-Short Baseline

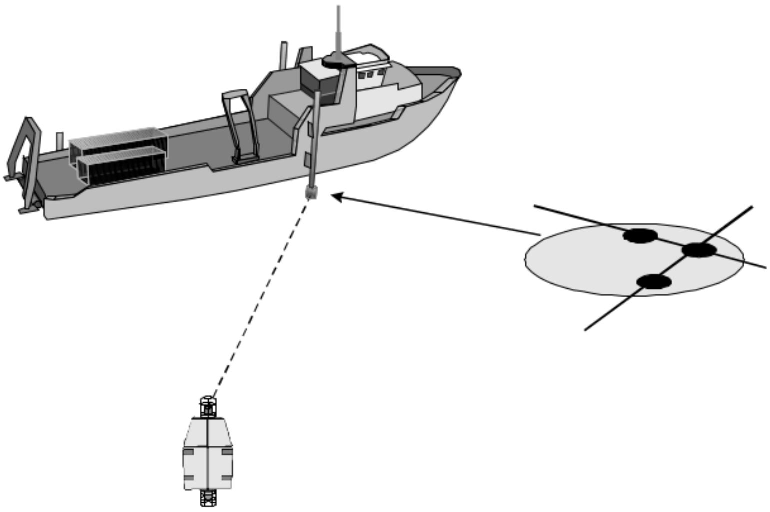

Figure 3.9: Ultra-Short Baseline - Source: Vickery

Ultra-Short Baseline (USBL) is a system of acoustic positioning which is similar to SBL where it uses an array of transceivers(three or more) and is typically fixed onto a surface vessel. The transponder is attached to underwater vehicle that is to be located (see figure 3.9). An acoustic pulse is transmitted by he transceiver and detected by the transponder on the underwater vehicle which replies with its own acoustic pulse. The returning pulse is detected by the surface vessel transceivers array. The time from the transmission of the initial acoustic pulse until the reply is detected, measured and converted into a range. The difference from the SBL system is that, instead of using trilateration to calculate a sub-sea position, the USBL system measures both range and angle from the underwater vehicle to the transceiver array. To avoid ambiguity in phase angles measurement, the transceivers in the array are typically separated only be half of the wavelength ( 10cm or less) of the acoustic signal. Hence, to determine the azimuth angle $\theta$, the phase difference of the signal between two receivers in the array is measured relative to the array's baseline. The angle is defined as the angle between the positive X-axis and the target position vector projected onto the horizontal XY plane. If a third receiver is used, orthogonal to the previous two, the elevation angle $\psi$ can be determined. The distance from the transceiver to the target r is the amplitude of the target vector. It is obtained by measuring the time of arrival as in LBL and SBL systems. The figure 3.10 and the Cartesian coordinate $(x,y,z)$ both given by Zhou [28] explains better the previous approach:

$$x = rsin\psi cos\theta \tag{3.1}$$

$$y = rsin\psi sin\theta \tag{3.2}$$

$$z = r\cos\psi \tag{3.3}$$

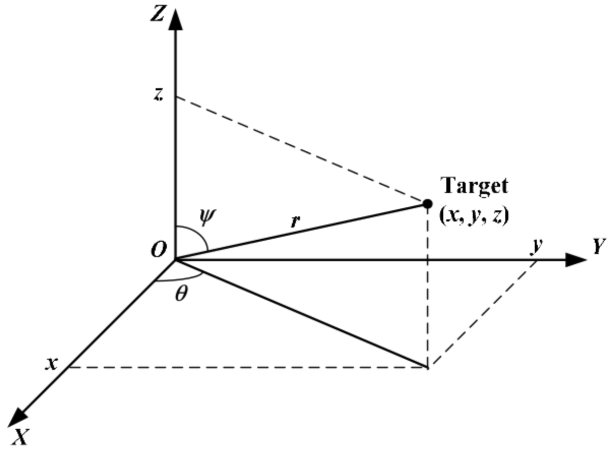


Figure 3.10: USBL Range and Angle Measurements - Source: Zhou

The transceivers in a USBL system usually are built into a single assembly in close proximity, which make USBL systems easier to be deployed. There is no need to deploy or calibrate the transponders array on the sea-floor. Since the range and the bearing measured in an USBL system is referenced to the transceivers mounted, additional sensors are needed to provide a position that is sea-floor reference. Giving this, the advantages of the USBL system are presented:

- Low system complexity makes USBL an easy tool to use.
- Ship based system - no need to deploys a transponder array on the sea-floor.
- Only a single transceiver at the surface - one pole/deployment machine.
- Good range accuracy with time of flight systems.

and the disadvantages:

- Detailed calibration of the system is required - usually no rigorously completed
- Absolute position accuracy depended on additional sensors - ship's gyro and vertical reference unit.
- Large transceiver/transducer gate valve or pole required with a high degree of repeatability of alignment

The typical sequence of event to determine the location of the underwater vehicle is as follows:

1. The USBL system emits a specific acoustic pulse to query the transponders in the vicinity.
2. The pulse travels through the water to the transponder.
3. The transponder detects the USBL signal and responds with a unique transponder acoustic pulse
4. The transponder pulse return through the water to the USBL array.
5. The USBL array detects the transponder signal and determines the round trip acoustic travel time and phase delay of the signal to each of the transducers in the USBL array.
6. The sound speed at the USBL array is used to calculate the received bearing and range of the transponder signal.

## 3.4 Conclusions

In this chapter we discussed the state of the art of autonomous docking system, where it was made an overview of the problem associated with docking (3.1), it was explained the difference between *docking* and *homing* (3.2) and it was shown a series of examples in literature about docking system that already exist and have been tested (3.2.1).

It was presented a brief explanation about Acoustic Positioning Systems (3.3) where it was presented the more used systems such has Long Baseline, Short Baseline and Ultra-Short Baseline. It was shown the advantages and disadvantages of each system in underwater positioning.

# Chapter 4

# Problem Statement

This chapter describes the docking problem between an AUV and an ASV. The concept in this section is influenced by methodology presented for Mobile Offshore Base (MOB) [29] by Girard, Sousa and Hedrick([30],[31] and [32]).

## 4.1 Introduction

In order to expand the AUV's autonomy it is important to have some sort of system that allows the vehicle to transfer power and data. This dissertation aims to develop a control strategy to allow a successful docking between an AUV and a ASV (see figure 4.1).

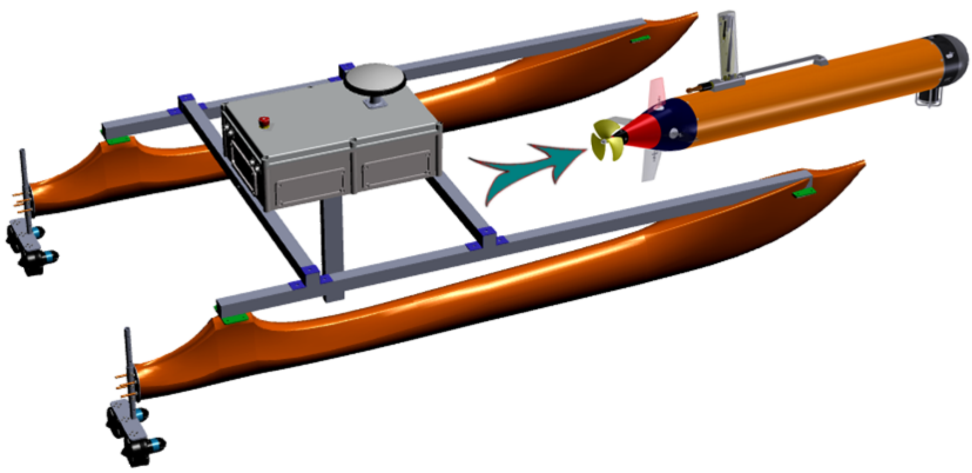

Figure 4.1: Docking between ASV and AUV

To achieve this goal the two systems are required to:

- The AUV must reach the position for docking defined by the operator.
- The ASV must find the AUV and proceed to the docking maneuver.

- After the docking maneuver, the vehicles will be mechanically connected to allow power and data transfer.

- The two vehicles may move together, under the control of the ASV, after the docking phase.

The following sections will present the system requirements for docking,the docking sequence, modes of operation to be considered in the approach and problems formulated to be solved in chapter 5.

## 4.2 Requirements

### 4.2.1 LAUV

The Light Autonomous Underwater Vehicle (LAUV)[33] designed and built at LSTS is a small underwater vehicle (see figure 4.2) optimized to be easily carried and have a huge amount of payload in a small mechanical structure. The main goal of developing a vehicle like this is to have the opportunity to test new software methodology without the concern of a high monetary impact in case of a catastrophic failure.


Figure 4.2: Autonomous Underwater Vehicle *Noptilus 2* - Source: LSTS

The LAUV is a torpedo shaped vehicle, with a length station at 110cm, a diameter of 15cm and a weight starting 18kg. The vehicle has an actuation system consisting of on propeller, and 4 direction control fins, all electrically driven. It uses a single board computer (SBC) running Linux that runs the control system software. For the navigation it uses an inertial measurement unit (IMU) and an attitude and heading reference system (AHRS). It is able to navigate with payload such has a Multibeam or Sidescan Sonar.

A more detailed specifications table is presented at 4.1.

| ***LAUV Specifications*** | |
|---|---|
| Length | Starting at 110 cm (depending on configuration) |
| Diameter | 15 cm |
| Weight | Starting at 18 kg |
| Endurance | Up to 8 hours @ 1.5 m/s |
| Wi-Fi | 2.4 GHz and/or 5 GHz |
| GSM/HSDPA | Quad-band 3G module |
| Maximum Depth | 100 meters |
| Inertial Navigation System | Maximum Gyro Bias: 1 degree per hour |
| Conductivity Temperature Depth (CTD) | Up to 6Hz sampling rate |
| Echo Sounder | Frequency: 675 kHz (Mounting pointing forward) |
| Camera | Resolution: 720p in H.264 and near 1080p in JPEG |
| Multibeam Sonar | Frequency: 260 kHz. Range: 100 meters |
| SideScan Sonar | Edgetech, Klein, or Imagenex |
| Satellite | Iridium SBD |
| Environmental Sensors | Crude and Refined Oils, Rhodamine, Chlorophyll |

Table 4.1: LAUV specifications

### 4.2.2 ASV

*Caravela* is an Autonomous Surface Vehicle developed in LSTS with the purpose of perform high-resolution bathymetry up river with a velocity of two meters per second using deltaT multibeam sonar(260 kHz) and high precision Global System Position (GPS). *Caravela* is a small catamaran-like vehicle with a length of 230cm, 145cm of width from one floater to another and a weight of 54kg (see figure 4.3).

Figure 4.3: Autonomous Surface Vehicle *Caravela* - Source: LSTS

The actuation system is based on 4 brushless motors electrically driven by 4 electronic speed controllers (ESC). It has an autonomy of 4 hours with the motors in full throttle or 10 hours with a mixed use. It has a payload of a Multibeam, Echosounder, USBL modem, 2 cameras for stereoscopic vision, differential GPS and a Wifi bullet for extended communication. To control all the payload and actuation, there is a SBC in use, that runs Linux, and controls all the system software.

| ***ASV Specifications*** | |
|---|---|
| Length | 230cm |
| Width | 145 cm |
| Weight | 54kg |
| Endurance | Up to 4 hours @ 2 m/s or up to 10h with mixed velocity |
| Wi-Fi | 2.4 GHz and/or 5 GHz |
| GSM/HSDPA | Quad-band 3G module |
| Inertial Navigation System | Maximum Gyro Bias: 1 degree per hour |
| Echo Sounder | Frequency: 675 kHz (Mounting pointing forward) |
| Camera | Resolution: Foscam FI9853EP Outdoor Waterproof H.264 720P PoE IP Camera |
| Multibeam Sonar | Frequency: 260 kHz. Range: 100 meters. deltaT |
| Satellite | Iridium SBD |

Table 4.2: ASV specifications

## 4.3 Models of the vehicles

This sections is intended to present the dynamic behavior models of the LAUV and ASV *Caravela*.

### 4.3.1 Linear model of LAUV

The linear equations of movement of the LAUV where obtained through the linearization of the expression in 2.13, being it around a break-even point or a time-variant reference according to a given reference

$$v_0(t) = [u_0(t)v_0(t)w_0(t)p_0(t)q_0(t)r_0(t)]^T \tag{4.1}$$

$$\eta(t) = [x_0(t)y_0(t)z_0(t)\sigma_0(t)\theta_0(t)\psi_0(t)] \tag{4.2}$$

The linearization is

$$M\Delta\dot{v} + \frac{C(v)v}{\delta v}\bigg|_{v_0}\Delta v + \frac{D(v)v}{\delta v}\bigg|_{v_0}\Delta v + \frac{\delta g(\eta)}{\delta\eta}\bigg|_{\eta_0}\Delta\eta = \Delta\tau_0 \tag{4.3}$$

where $\Delta v(t) = v(t) - v_0(t), \Delta\eta(t) = \eta(t) - \eta_0(t)$ and $\Delta\tau(t) = \tau(t) - \tau_0(t)$.

The six DOF equations system can be decoupled into three independent subsystems, in order to facilitate the control of the vehicle movement [24] [34]. Therefore, the three subsystems and its state variables are

- Speed System - $u(t)$
- Steering System - $v(t)$,$r(t)$ and $\sigma(t)$
- Diving System - $w(t)$, $q(t)$, $\theta(t)$ and $z(t)$

The LAUV configuration suggests that the speed system can be controlled by the DC motor rotation, the steering system be controller by the vertical fins and the diving system controlled by the horizontal fins. The SISO (single-input single-output) simplification makes easier to apply the control architecture of the vehicle. The three subsystems equations are

**Speed System** - neglecting the interactions of *sway*, *heave*, *roll*, *pitch* and *yaw*, the speed equation is

$$(m - X_{\dot{u}})\dot{u} = X_{|u|u}|u|u + X_{prop} \tag{4.4}$$

It is assumed that the quadratic damping is the dominant dissipative effect.

**Steering System** - assuming that the velocity component is steady state are $v_0 = w_0 = p_0 = q_0 = r_0 = 0$ and $u_0 \neq 0$ and the break-even point is defined by $\sigma_0 = \theta_0 = 0$, the equation system is

$$\begin{bmatrix} m - Y_{\dot{v}} & mx_G - Y\dot{r} & 0 \\ mx_G - N_{\dot{v}} & I_{ZZ} - N_{\dot{r}} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{v} \\ \dot{r} \\ \dot{\psi} \end{bmatrix} + \begin{bmatrix} -Y_v & mu_0 - Y_r & 0 \\ -N_v & mx_G u_0 - N_r & 0 \\ 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} v \\ r \\ \psi \end{bmatrix} = \begin{bmatrix} Y_\delta \\ N_\delta \\ 0 \end{bmatrix} \delta_r \tag{4.5}$$

where $Y_\delta = Y_{uu\delta r} u_0 u_0$ and $N_\delta = N_{uu\delta r} u_0 u_0$.

**Diving System** - assuming that the vehicle moves at a constant *surge* $(u_0 \neq 0)$ speed and $\theta_0 = 0$, the speeds assume a value of $v_0 = p_0 = r_0 = 0$ and the vehicle works on the break-even point defined by $\sigma_0 = \psi_0 = 0$. A simplification can be made being $x_G = 0$ and the speed in $z$ (*heave*) is small when the vehicle is descending ($w = 0$). The linear model is

$$\begin{bmatrix} \dot{q} \\ \dot{\theta} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} \frac{M_q}{I_{yy} - M_{\dot{q}}} & -\frac{(z_g - z_b)W}{I_{yy} - M_{\dot{q}}} & 0 \\ 1 & 0 & 0 \\ 0 & -u_0 & 0 \end{bmatrix} \begin{bmatrix} q \\ \theta \\ z \end{bmatrix} + \begin{bmatrix} \frac{M_\delta}{I_{yy} - M_{\dot{q}}} \\ 0 \\ 0 \end{bmatrix} \delta_s \tag{4.6}$$

where $M_\delta = M_{uu\delta s} u_0 u_0$.

The steering and diving linear models represent an approximation of the model of the vehicle when it is actuated in a decoupled way in the vertical and horizontal planes where the vehicle keeps it break-even state.

### 4.3.2 Model of ASV

Typically the models for the surface vehicles are simpler compared to the underwater vehicles. In this case, is common to assume that *heave*, *pitch* and *roll* are negligible being the case that the vehicle only moves in the horizontal plane, at the surface of the water. Hence, the vectors that represent the state variables are reduced to $R^3$

$$\eta = [xy\psi]^T \tag{4.7}$$

$$\eta = [uvr]^T \tag{4.8}$$

The origin of the vehicle reference is chosen in the central part of it, $y_G = 0$. The mass of the vehicle must be evenly distributed a should present a symmetry in the $xz$ plane.

#### 4.3.2.1 Non-linear model for the ASV

The linear model proposed by Blanke [35] is a simplification of the model presented in 2.12 which is

$$(m - X_{\dot{u}}\dot{u}) = X_{|u|u}|u|u + (m + X_{vr})vr + (mx_G + X_{rr})rr + \tau_u \tag{4.9}$$

$$(m - Y_{\dot{v}}\dot{v}) + (mx_G - Y_{\dot{r}})\dot{r} = -(m - Y_{ur})ur + Y_{uv}rv + Y_{|v|v}|v|v + Y_{|v|r}|v|r + \tau_v \tag{4.10}$$

$$(mx_G - N_{\dot{v}})\dot{v} + (I_{zz} - N_{\dot{r}})\dot{r}) = -(mx_G - N_{ur})ur + N_{uv}uv + N_{|v|v}|v|v + N_{|v|r}|v|r + \tau_r \tag{4.11}$$

where $\tau$ is the ASV propulsion.

Assuming that the vehicle only moves in the horizontal plane, a simplification on the Kinematics equation, presented in 2.14, which is

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\psi} \end{bmatrix} \begin{bmatrix} cos(\psi) & -sin(\psi) & 0 \\ sin(\psi) & cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ v \\ r \end{bmatrix} \tag{4.12}$$

**Control and Propulsion System of ASV *Caravela*** - Usually, a surface vehicle has a propulsion system and a rudder, responsible for controlling the vehicle direction. In the case of *Caravela*, the propulsion is made by four thrusters fixed to back of the floaters. Despite this, the propulsion is made as if it only had one thruster on each side.

The force applied by each thruster, left and right, is $T_l(n_l, u)$ and $T_r(n_r, u)$, respectively and is defined by

$$T(n, V_a) = T_{|n|n}|n|n + T_{|n|V_a}|n|V_a \tag{4.13}$$

where $T_{|n|n} > 0$ and $T_{|n|V_a} < 0$ are the coefficients of the model based on the SNAME notation [10], $n$ is the coupling rotation speed and $V_a$ is the forward speed of the thruster.

Then, the force and torque vectors ($\tau$) of the thrusters are

$$\tau = \begin{bmatrix} \tau_u \\ \tau_v \\ \tau_r \end{bmatrix} = \begin{bmatrix} (T_r + T_l) \\ 0 \\ (T_l - T_r) r_f \end{bmatrix} \tag{4.14}$$

The rotation of the propeller on each coupling will apply a force to the vehicle on the $x$ plane. Since this force is applied outside of the longitudinal line of the vehicle (i.e. $y \neq 0$), it will also cause torque proportional to the force and distance in the $yy$ axis. Considering that the propulsion force is the sum of all the forces of each thruster, this will be designated *common thrust* ($T_{com}$). On the other hand, being the applied torque defined by the difference between the applied forces of the thrusters, this will be the *differential thrust* ($T_{diff}$).

#### 4.3.2.2 Linear model for the ASV

Applying the same procedure has in 4.3.1, it is possible to obtain the linear movement equations of the vehicle.

**Speed System** - Considering that *caravela* has long floaters, it is normal to get some sort of inertia when it is performing a rotation. Being this, the rotation speed $r$ will have always small values where $(mx_G + X_{rr})rr \approx 0$. Also, the cross-speed $v$ will also be considered small where $(m + X_{vr})vr \approx 0$.

The speed equation will be

$$(m - X_{\dot{u}})\dot{u} = X_{|u|u}|u|u + \tau_u \tag{4.15}$$

**Steering System** - applying a lineatization around $u_0$ with $v_0 = r_0 = 0$, the system takes a similar form has the model in Davidson and Schiff [36] where

$$\begin{bmatrix} m - Y_v & mx_G - Y_{\dot{r}} \\ mx_G - N_{\dot{v}} & I_{ZZ} - N_{\dot{r}} \end{bmatrix} \begin{bmatrix} \dot{v} \\ \dot{r} \end{bmatrix} + \begin{bmatrix} -Y_v & mu_0 - Y_r \\ -N_v & mx_G u_0 - N_r \end{bmatrix} \begin{bmatrix} v \\ r \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tau_r \tag{4.16}$$

and

$$\begin{bmatrix} \dot{v} \\ \dot{r} \end{bmatrix} = M^{-1} N(u_0) \begin{bmatrix} v \\ r \end{bmatrix} + M^{-1} b \tau_r \tag{4.17}$$

An alternative to the previous model is the Nomoto's models [37]. Eliminating the cross-speed $v$ in 4.17, the second order transfer function of Nomoto's, between $r$ and $\tau_r$ is

$$\frac{r}{\tau_s} = \frac{K(1 + T_3 s)}{(1 + T_1 s)(1 + T_2 s)} \tag{4.18}$$

Since the time constant is given by $T = T_1 + T_2 - T_3$, a first order equation approximation of the Nomoto's model can be

$$\frac{r}{\tau_r}(s) = \frac{K}{(1+T_s)} \tag{4.19}$$

## 4.4 Modes of Operation

To have a successful docking performance between the ASV and the AUV, one must define a proper set of **modes of operation**:

- **Unassembled mode** - the ASV and the AUV are two independent modules, with different position and orientation, and different mission assignments, defined prior to the docking maneuver.
- **Homing mode** - this mode is different to both of the vehicle where:
    1. The AUV will go to a point defined by the operator to receive the docking from the ASV, staying in that point within a defined radius,
    2. The ASV will perform the homing sequence after the AUV is in the docking zone, will do collision avoidance in order to perform the docking with the risk of frontal or lateral collision between the two vehicles.
- **Docking mode** - the ASV is performing a motion plan to dock above the AUV
- **Docked mode** - the AUV is docked in the ASV
- **Abort mode** - set of maneuvers that allow the ASV to retrieve or retry the docking maneuver.

An existing problem when the vehicles are performing the *unassembled* or *homing mode* is the possibility of collision between both, leading to serious malfunctions. Therefore, the docking system must be capable to avoid collisions between both vehicles.

During *docking mode*, there is the necessity to have a controller that track the motion plan the drives the ASV to the AUV, to perform docking. Thus, external disturbances, i.e. ocean currents, need to be taken into to account in order to build a robust and safe controllers. During the final stage of docking, there is the possibility that the external disturbance can drive the LAUV or the ASV way from the motion plan and add a risk of collision to the performing maneuver. In this case, the control must be able to stop the docking maneuver and drive the ASV away from the LAUV - *abort mode*.

## 4.5 Assumptions

This section introduces some **assumptions** that will be a remainder of the problems formulated in the following sections.

- The following problems will be formulated in the horizontal plane only (XY) given that the docking maneuver will be performed at the surface of the water and the following state variables will be considered for the ASV and AUV:
  1. Geodetic position frame - Latitude and Longitude,
  2. Earth-fixed frame - x and y,
  3. Heading - $\psi$
- The ASV knows its exact position and orientation, related to the geodetic frame and the earth-fixed reference frame.
- The ASV knows the exact position and orientation of the AUV at a given time.
- The ASV must approach the AUV from behind, i.e., if one considers that the ASV position is expressed in the AUV body-fixed reference frame, then the docking point of the two vehicles is in the origin of the reference frame, aligned with the x-axes. Therefore the ASV must perform the motion plan in order to reach the docking point, along the negative x-axes. This assumption can be seen in figure 4.4.

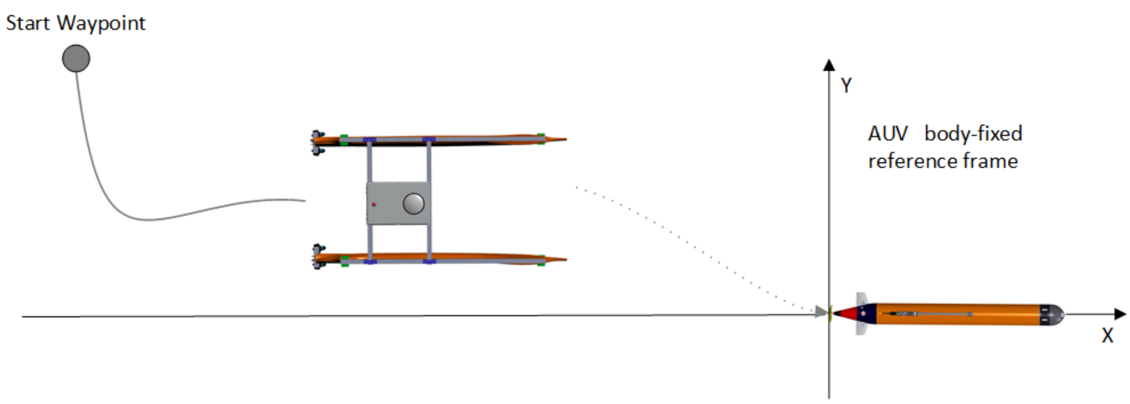


Figure 4.4: Docking problem expressed in the AUV body-fixed reference frame

## 4.6 Problem 1 - Docking Maneuver Controller

This problem intends to develop a operator command strategy, based on the LSTS Toolchain (2.4). The idea is to the operator launch the docking maneuver for both of the vehicles, through an interface, and monitor in real-time the actions being done by them.

### 4.6.1 Problem description

So that could be possible to have a proper execution of the docking maneuver it is necessary to have an interface between the operators and the vehicles.

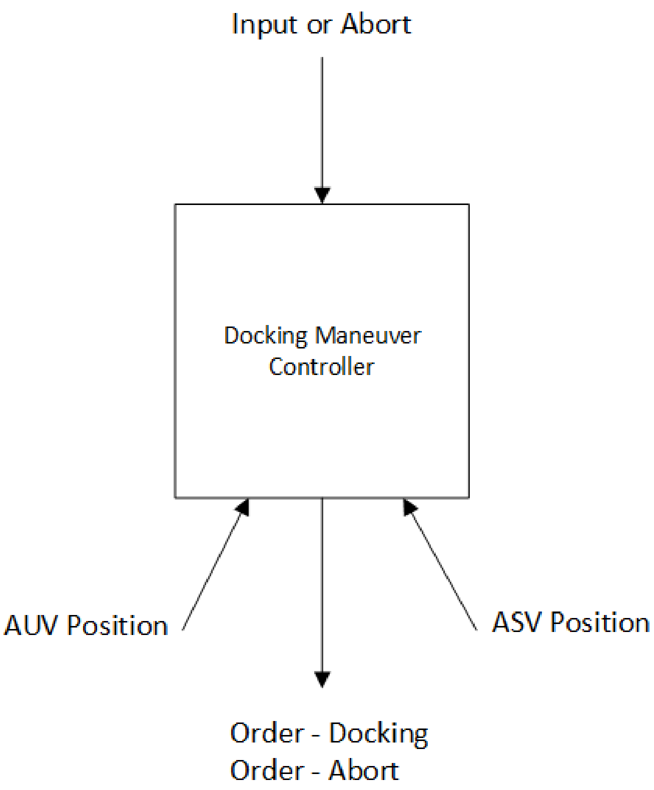


Figure 4.5: Docking Maneuver Controller Diagram

Figure 4.5 presents a diagram of the interaction between the interface and the operator/vehicles. The operator would insert inputs to the interface, such has point to dock and what vehicle should perform the maneuver. It would receive the vehicles position, to have a real-time information about the whereabout of the vehicles.

## 4.7 Problem 2 - Docking Supervisor (ASV)

The purpose of this problem is to have a supervisor that would change the phases of the docking maneuver seen in 5.1 based on the trigger events proposed.

### 4.7.1 Problem description

This controller will verify the events triggered from the docking maneuver. With this, it will adjust the maneuvers phases. It will exchange information between the 2 vehicles, using a communication protocol, which will send messages pointing, for example, the availability for docking on both vehicles, the need for position correction or the maneuver abortion. The figure 4.6 shows the diagram of the operation of the supervisor.

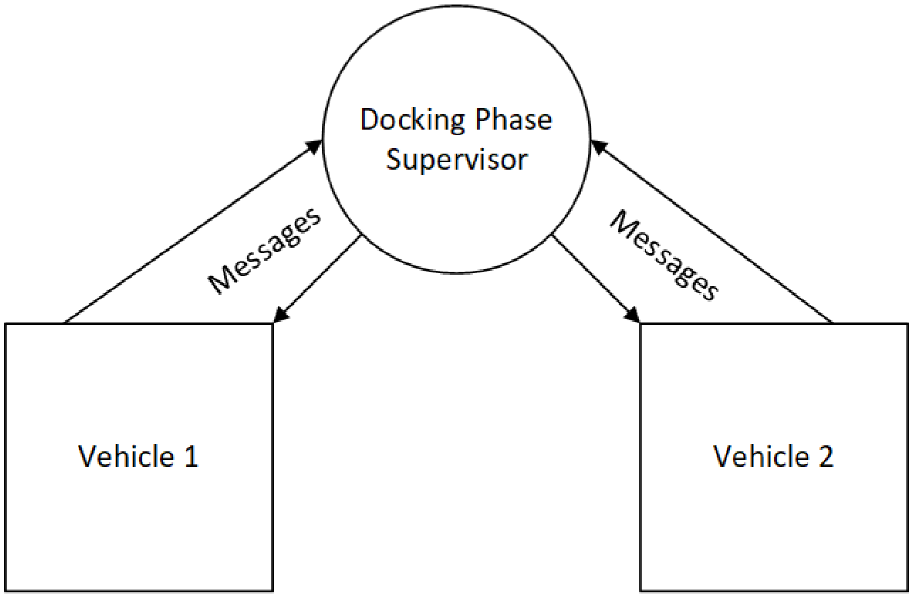


Figure 4.6: Docking Phase Supervisor Diagram

## 4.8 Problem 3 - Vehicle Position Correction Maneuver with External Disturbances (ASV and AUV)

Considering that the vehicles when immobilized at the surface of the water, they can be subjected to external disturbances, it is necessary to add a sub-maneuver that will correct the position of the vehicle. This waypoint should be defined by the operator or by the supervisor controller on the actual position of the vehicle.

### 4.8.1 Problem Description

This problem intents to find a way to maneuver the vehicle when it is outside the circle defined around a defined point and correct its position.

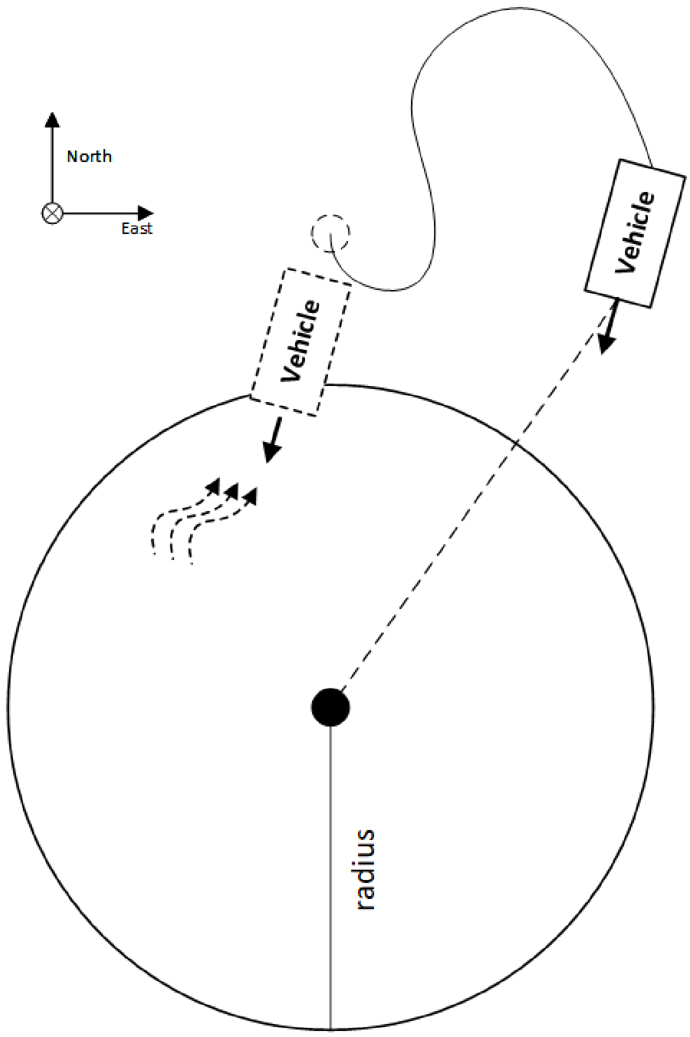


Figure 4.7: Docking Phase Supervisor Diagram

The figure 4.7 shows a representation of this maneuver, where the vehicle would go to the waypoint defined. Then a virtual circle would be created around the point. When the vehicle is inside this circle it is considered to be in a stationary mode. When the vehicle leaves the circle, it should return to the waypoint previously defined. It will be necessary to have into account that after the vehicle makes the position correction, the actual heading could be different from the previous one and must be reported in order to make the necessary corrections.

## 4.9 Problem 4 - Move to waypoint Maneuver (ASV and AUV)

### 4.9.1 Problem Description

This problem intents to add a functional way to make the vehicle move from one point to another. This point should be calculated by the controllers in order to make the vehicle correct it's heading and position. Figure 4.8 shows a simple diagram how this maneuver would work.

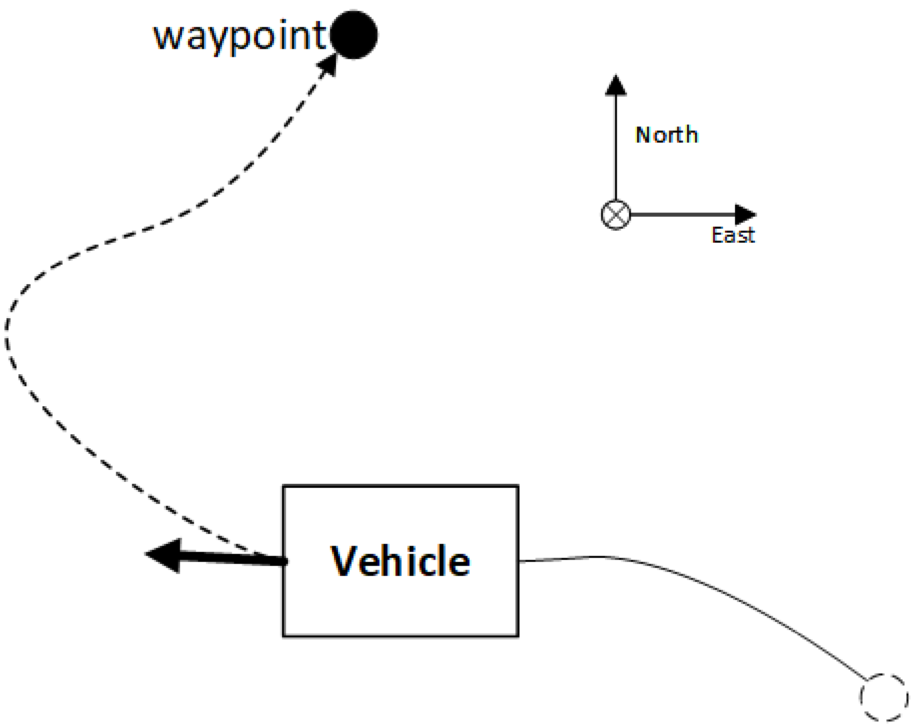


Figure 4.8: Move to Waypoint Maneuver Diagram

## 4.10 Problem 5 - Homing of the vehicle (ASV)

As described before, in the literature there two approaches to the docking maneuver where the first can be only the docking to the dock station and the second uses first the homing maneuver and then the docking maneuver.

### 4.10.1 Problem Description

The problem intents to apply the second approach.

The idea would be that the vehicle that functions has the docking station would move to the periphery of the vehicle that is to be docked. Figure 4.9 shows an example of the approach of homing where vehicle 1 would the docking station and vehicle 2, the one to be docked.

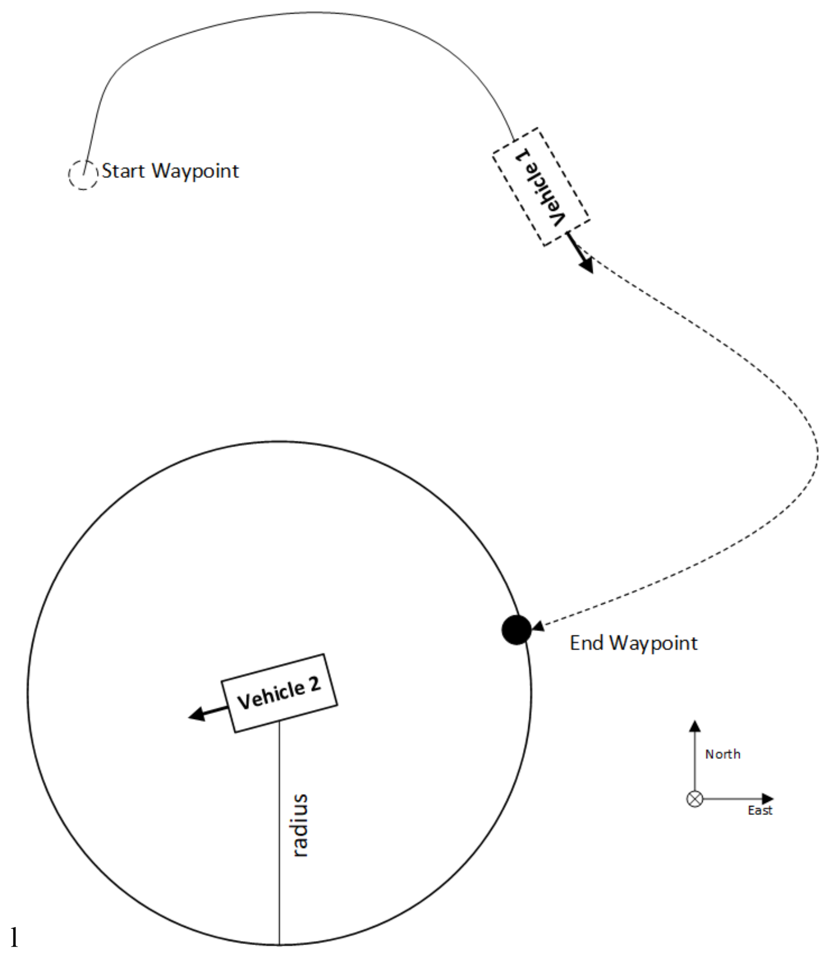


Figure 4.9: Homing Diagram

Vehicle 1 would leave the station keeping maneuver, when vehicle 2 indicates that is ready to perform the docking in the desired waypoint. This would move until a waypoint defined by a circle with a radius, around the actual position of vehicle 2. The waypoint would be defined in a way that it would be at the distance of the radius of vehicle 2 position and at 180º of the heading of the former one. This would guarantee that both of the vehicle would be with the same heading when entering the docking phase.

## 4.11 Problem 6 - Docking of the vehicles (ASV)

### 4.11.1 Problem Description

This problem intents to give continuity to the docking maneuver between the two vehicles. As seen in figure 4.10 after finished the homing maneuver with success it is necessary to change the approach so that it is possible to conclude the docking maneuver between the two vehicles with success.

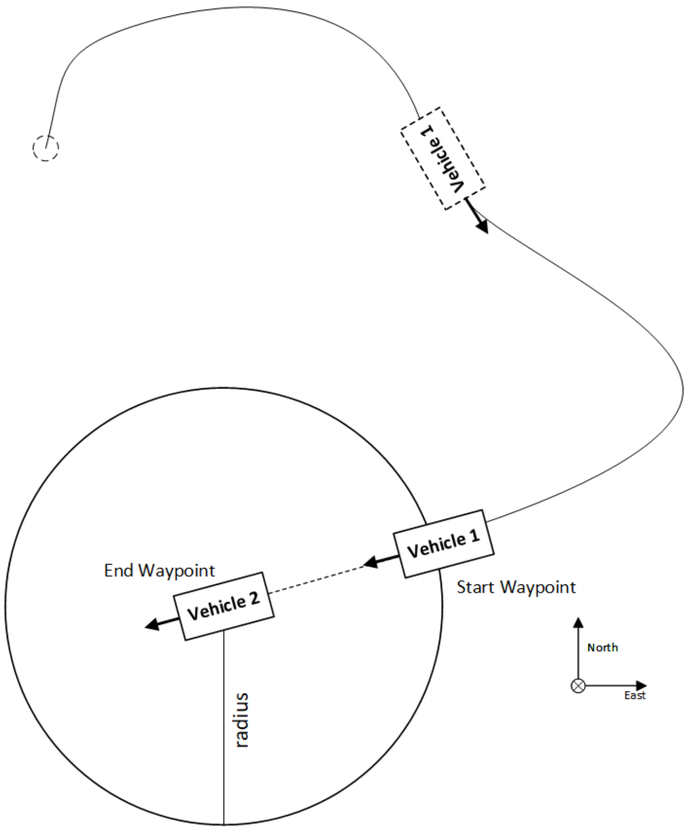


Figure 4.10: Docking Diagram

After vehicle 1 being in the homing point and with the correct heading, it will move towards vehicle 2 reducing it's velocity as the the distance between the two is reduced. When the vehicle 1 is with a distance from vehicle inferior to 5 meters and the docked trigger is active, the supervisor should change to the phase of docked.

## 4.12 Problem 7 - Abort sequence (ASV and AUV)

The abort sequence should be performed in at least two situations which are, when the vehicle detects an imminent collision between the 2 vehicles given their position and when the docking attempt was failed.

### 4.12.1 Problem Description

This problem intents to make an approach to the abort sequence given the two previous situations. When the vehicle is performing the homing or the docking, the supervisor must detect the actual position of both vehicles and make them diverge from each other when a potential collision is detected. If the case is when the docking is attempted but failed, the supervisor, based on the information given by the operator, must retry the all docking sequence again, if it is the case, or abort the maneuver.

## 4.13 Conclusions

In this chapter we presented the requirements of the vehicles to be used in the docking approach which is a LAUV (4.2.1) and the ASV *Caravela* (4.2.2).

It was presented the model of each vehicle where it was presented the linear model of the LAUV (4.3.1) and for the ASV it was shown the non linear model of the vehicle (4.3.2.1) and next, the linear model (4.3.1).

The modes of operation where shown in (4.4) where it is displayed the docking sequence to be implemented.

Following it was presented the assumptions (4.5) to be considered finally the problems to be taken into account where the first one is the Docking Maneuver Controller (4.6) which gives the operator a way to communicate with the vehicles, Docking Phases Supervisor (4.7) which controls the phase state of the docking sequence presented in 4.4 and is trigger by events defined in the section.

It is defined a Vehicle Position Correction Maneuver of External Disturbances (4.8) and the maneuver to move the vehicles from one point to another in Move to waypoint Maneuver (4.9).

The Homing of the vehicle (4.10) is shown to place the vehicle in a proper position to make the docking approach and the Docking of the vehicles (4.11) where is made the final approximation of both vehicles.

It is described why it is needed an Abort Sequence (4.12).

# Chapter 5

# Approach

This section presents the approach to the problems presented in the previous chapter. It is based on the application of a docking solution using the toolchain existing in the LSTS and to be used on real physical vehicles such has the ASV *Caravela* and a proper AUV.

## 5.1 Docking Sequence

In order to have a successful docking maneuver and taking into account that for that it is necessary to managed the cooperation between two independent systems, a docking sequence diagram was made. The figure 5.1 shows a example to be applied in the approach.

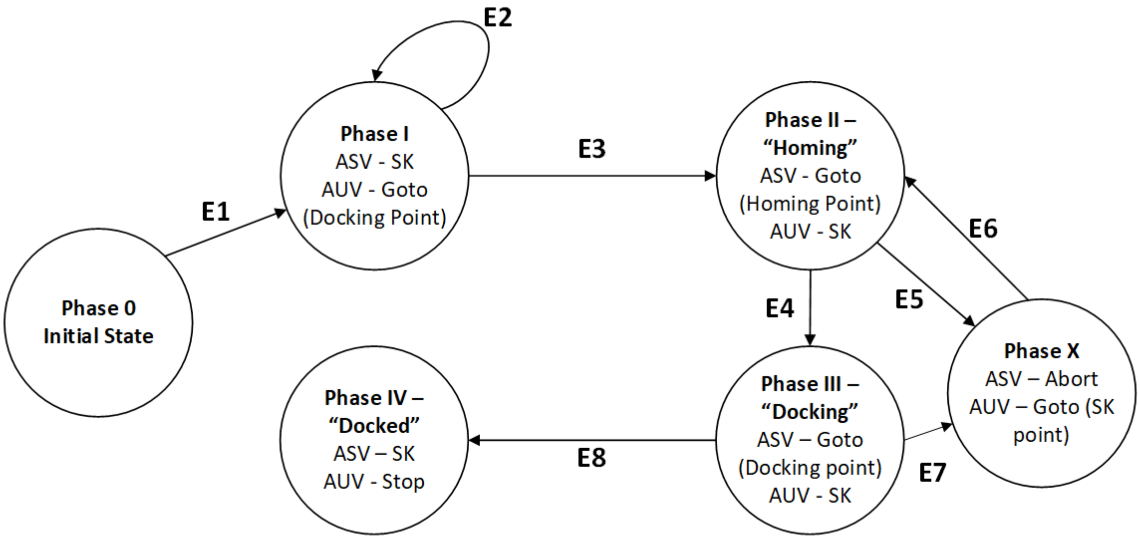


Figure 5.1: Docking sequence

The different phases presented, represent states of the docking maneuver in all of the vehicles. This states will be managed by a supervisor, described previously in *Problem 2 - Docking Supervisor (ASV)* (4.7). The supervisor will receive information from low-level tasks and manage the states based on trigger events.

### 5.1.1 Docking phases

This phases will assign functions to each vehicle based on each function in the docking maneuver. They comprise as follows:

- **Phase 0 "Initial State"** - this phase represents the state of the vehicles before they initiate the docking maneuver. They may be executing other maneuvers or in a "hold state". Both systems are decoupled from each other.
- **Phase I** - in this phase, the docking maneuver is initiated after a trigger event, that could be from the operator or from a previous maneuver plan. This phase will place the ASV in Station Keeping (SK), meaning, it will stay in the actual position correcting it when it gets driven by external disturbances such has ocean currents. This is the solution to *Problem 3 - Vehicle Position Correction Maneuver with External Disturbances (ASV and AUV)* (4.8) in the previous chapter and will be detailed later in this chapter. The AUV will move to the position defined has the docking point, using the Goto maneuver, which is described in *Problem 4 - Move to waypoint Maneuver (ASV and AUV)* (4.9) which will also be described in more detail later.
- **Phase II "Homing"** - in this phase, after the ASV got the confirmation from the AUV that the latter is available to receive the docking, which means it is on the defined point of docking, it will move to a position in the vicinity of the AUV. This position will be calculated based on the information gathered from the USBL modem. The AUV will execute the maneuver of station keeping (SK) in the defined point in order to perform the docking maneuver. This maneuver will verify the relative position between the two vehicles and will take measures so that both vehicles have the same heading at the end of the maneuver.
- **Phase III "Docking"** - this is posterior to *Phase II "Homing"* and will take the ASV near the AUV in order to make the final approximation a dock with the latter. The ASV will slow down as the distance between the two vehicle will be reduced. It will stop this approximation when both vehicle signal that the docking has been accomplished.
- **Phase IV "Docked"** - this phase indicates that the docking was completed and that the ASV is docked with the AUV. After this, the ASV takes control of both systems, where the AUV will stay on "hold" until further notice, and will execute a SK or take the AUV to another position.
- **Phase X** - this is a phase that is initiated when the ASV is executing *Phase II "Homing"* or *Phase III "Docking"* and the AUV needs to correct its position . Here, the AUV will move to the SK point and the ASV will abort its actual maneuver. After the AUV is again in position, if the ASV was doing the homing phase, it will start it again but if it was executing

the docking phase, it will start with the homing phase in order to correct its heading based on the new pose of the AUV.

### 5.1.2 Event trigger

The phases will change based on the a series of triggers of events. This event triggers in the docking sequence are presented in table 5.1:

| Event | Condition |
|---|---|
| E1 | Start docking maneuver? |
| E2,E5,E7 | SK radius >5m? |
| E3 | AUV ready for docking? |
| E4 | Range between vehicles <50m? AND difference of heading = 0? |
| E6 | AUV on SK point? |
| E8 | Docked trigger? |

Table 5.1: Event trigger conditions

They will trigger the phases in the docking sequence. Each events represents a sort of "questions" that are made by the supervisor controller (4.7). Each event is explained as follows:

- E1 - this asks if the operator gave the order to start the docking maneuver or it was triggered by the ending of another maneuver int the vehicle plan.
- E2, E5, E7 - SK means *Station Keeping* which is the way to maintain the vehicle in a given area when there are external disturbances actuating on it. This event makes the question if the actual position of the vehicle is 5 meters far from the point defined for the SK. If affirmative, the vehicle should return to the former position.
- E3 - this event is triggered when the AUV reach the position defined and is ready to perform the docking. It give this information to the supervisor controller (4.7).
- E4 - when the vehicle reaches the homing position the event will indicate to the supervisor that is ready to perform the docking approach. This will mean that the range between the two vehicles is < 50 meters and they have the same heading.
- E6 - this event is triggered after the vehicle reaches again the SK position.
- E8 - when the vehicle reaches the other vehicle, it will trigger that has been docked and finish the docking maneuver.

## 5.2 Docking toolchain approach

As stated before in section 2.4, LSTS designs and builds autonomous vehicles and it too develops software ranging from operator interface (Neptus - 2.4.1) to the interaction with sensor's firmware (DUNE - 2.4.3) and all the communications in between (IMC - 2.4.2).

Considering the toolchain architecture, the approach to the problem will be done as follows:

- **DUNE** - Tasks that receive information like, for example, vehicle location, docking availability, fail and abort flags.
- **IMC** - define messages to establish communication between the interface *Neptus* and *Dune* or between *Dune* running on different vehicles.
- **Neptus** - maneuver with relevant parameters to the docking maneuver.

Since the approach involves the synchronization between two vehicles and the operator, it is necessary to build diagrams based on the *Hybrid Systems Theory* (2.3) so that can exist a good interaction between the vehicles without incurring in faults or dead-locks.

### 5.2.1 Inter-Module Communication approach

So that could be possible to perform communication between low level tasks, has seen in DUNE, to high-level operator interactions, has seen in Neptus, it was necessary to create communication messages based on the IMC protocol, in order to fulfill this need:

1. **Docking**.
2. **DockingState**.
3. **DockingUSBL**.

#### 5.2.1.1 *Docking* message

This messages serves has a connection between Neptus and the vehicles that will perform the docking maneuver. It has the following parameters:

- Vehicle function.
- Vehicle target.
- Number of retries.
- Maximum speed.
- Speed Units.
- Latitude (WGS-84).
- Longitude (WGS-84).

**Vehicle function** - defines which vehicle will perform or receive the docking maneuver. The parameters can be *target* or *station* where the vehicle that has the first parameter will be the one to receive the docking maneuver and the vehicle with the *station* parameter will be the one to perform the docking. This serves has a way to the same task run in different ways in different vehicles. For example, assuming that it is intended that an AUV (i.e *xplore-1*) will receive docking and the ASV (*Caravela*) will perform docking, when the operator is planning the Neptus plan, he will assign the *xplore-1* with the vehicle function of *target* and the Caravela with *station* function.

**Vehicle target** - this parameter define the target of the vehicle in order to perform docking. It exists so that the low level task can identify the target vehicle based on its system id such has for example *xplore-1* or *caravela*. Taking the example in **Vehicle Function**, in the case of the AUV, the target would be *caravela* and vice-versa. This creates a way to the lower tasks discriminate the receiving messages in order to use only the one that matters to the functioning of the maneuver.

**Number of retries** - this ensures that the operator can choose between not retry the docking maneuver in case of failure or retry in the case it is intended.

**Latitude and Longitude (WGS-84)** - this will give the lower level task the position where to perform the docking. This will only be used by the vehicle that has it's function has *target*.

#### 5.2.1.2 *DockingState* message

This message emerged in the need to have an exchange of information between two vehicles. It shows the availability of both vehicles to perform docking and how is the state of the docking maneuver. It consists of the following parameters:

- System name.
- State.
- Availability.
- Vehicle Function.

**System name** - this parameter shows the system name from where the message was generated. Since this message will be sent both ways through the vehicles, it was necessary to add an identifier in order to the tasks know where the message is coming from.

**State** - since we are performing a docking maneuver, it was necessary to add this parameter to the message, where there is the indication when the docking was successful or not. When it is successful, the flag *Docked – Success* is chosen to be added to the *State* parameter. On the other hand, when there was a failure in the docking attempt, the flag will be *Docked – Failure*. This serves has an indication from one vehicle to another that the docking was successfully done or not.

**Availability** - this is an important parameter when performing docking. It serves has an indicator that there is a possibility to perform docking or not. When a vehicle is ready to perform or receive the docking, the parameter will have the flag *Ready* giving the indication that it is possible to perform the docking. When the attempt of docking is being done, the parameter will have the flag *Performing* having it changed to *Abort* only when something fails in the attempt. With this parameter it is possible to stop a docking maneuver attempt when, for example, one of the vehicles need to correct its position.

**Vehicle Function** - this parameter serves has a way to test if there is no more than one vehicle announcing a function on the bus. It will be verified and if two vehicles have the same function, they will stop the docking attempt and return an error, announcing that there is multiple vehicles with the same function.

#### 5.2.1.3 *DockingUSBL* message

This message will be used in the simulator *DockingUSBL* which simulated the ranging and bearing of two vehicle through an USBL modem. It will be used to encapsulate a *EstimatedState* message, so that could be possible to calculate the range and bearing between the two vehicles. It will be sent in the AUV and receive in *Caravela*, in the same task *DockingUSBL*. This essencially serves has a way to send a specific *EstimatedState* message inside the IMC bus of the AUV so that it would not be confused with other *EstimatedState* messages and give wrong information to vital task operating the vehicle. The task Docking USBL after receiving this specific *EstimatedState* message will fill the *USBLAnglesExtended* message with the angle between the two vehicles.

#### 5.2.1.4 Message exchange between two vehicles

The LSTS toolchain in general, has as structure some how similar to a master/slave relationship. This means that the information exchanged is only between the vehicles (DUNE) and Neptus. Communication between independent vehicles is not used. However, since the docking process is complex it was decided that to implement a way to having the two vehicles exchanging messages. This makes the docking process easier to be implemented, since the supervisor of the maneuver do not need to be on the higher level (Neptus) but can be implemented on one of the vehicles, i.e. *Caravela*. Thus, since *Caravela* is normally the docking station and consequently will be the one to move to the AUV, it is faster to make corrections of it's position without have a third party causing a delay on the communications. This communication is made through the use of the Transmission Control Protocol (TCP).

### 5.2.2 Neptus approach

Given that it is necessary to have a platform that possibility the launch of docking maneuvers in the LSTS vehicles, it is necessary to add this maneuvers in Neptus software. This would be, in the planning phase, to launch the maneuver in a single way or with other maneuvers. So, this section will show the approach done when creating a maneuver in Neptus software.

#### 5.2.2.1 Problem 1 - Docking Maneuver Controller

In order to have some sort of control over the vehicle before and after the beginning of the docking maneuver it was needed to have an interface between the operator and the vehicles. This was made using Neptus software (2.4.1) which is an *Command and Control* software built in LSTS. It uses a simple interface where it is possible to see the vehicles location in real time, launch plans with multiple maneuvers and review previous missions.

Figure 5.2 shows a simplified diagram of the sequence to be performed when adding the docking maneuver in Neptus.

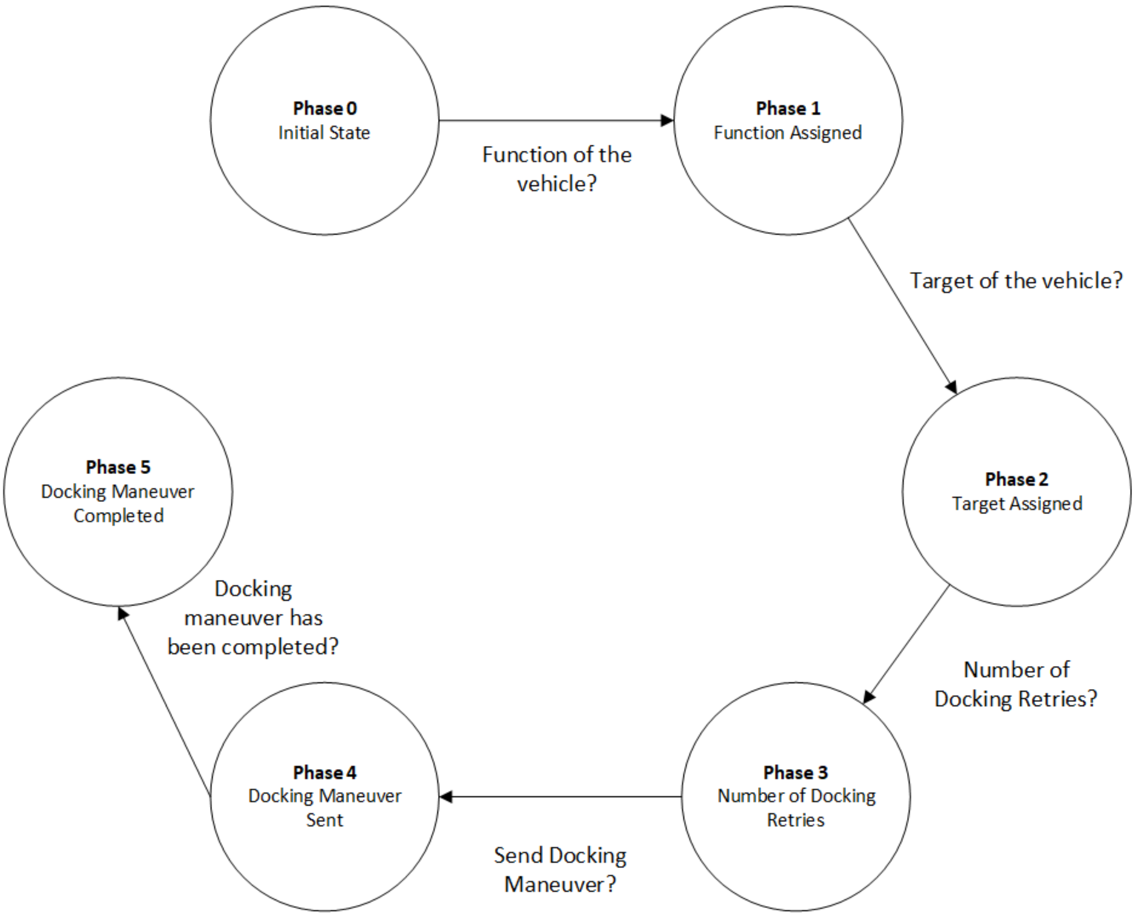


Figure 5.2: Neptus Sequence Diagram

Here, the operator when adding the docking maneuver to the interface must go through the following phases:

- **Phase 0** - this phase is the initial state, when appears the menu to the operator to inserted the needed information to be sent to the low-level tasks.
- **Phase 1** - in here, the function of the vehicle is asked. In this case, the operator has two functions that he can assign to the vehicle:
  1. *Target* - where this vehicle will be the one to receive the docking from the station
  2. *Station* - where this is the vehicle that will perform the docking maneuver.

  This will enable that the same maneuver will work on both vehicle but with different outcomes given the assigned function.
- **Phase 2** - here, the target of the vehicle will be the one that has the opposing function of the actual vehicle. What this means is that, for example, if the ASV is assigned with the function of "Station", its target will be the AUV that has the function "Target" and vice-versa. This enables the low-level task to filter the messages that exist on the IMC(2.4.2) bus and only use the ones that refer to the target of the maneuver.
- **Phase 3** - this phase asks for the number or retries of the docking maneuver. This retries are done when the vehicles fails to perform the docking maneuver.
- **Phase 4** - this phase happens when the maneuver is sent to be performed by the vehicles.
- **Phase 5** - this phase is active when the vehicles have successfully completed the docking maneuver.

#### 5.2.2.2 Docking maneuver specifications

As explained in 5.2.1 and presented in 5.2.2.1, the maneuver made in Neptus software will send relevant information to the vehicles through the IMC protocol. This information is necessary for the good functioning of the maneuver and consists in:

1. Location
   - Location
   - Z
   - Z units
2. Docking specifications
   - Docking Target
   - Number of Retries

- Vehicle Function

In **Location**, it is indicated the place to perform the docking maneuver, based on ECEF coordinates and a what depth is must be done, where in this case is zero meters. This **Location** parameter is only important for the vehicle for the vehicle that will receive the docking.

The **Docking Specification**, has the following parameters:

**Docking Target** - where it is placed the vehicle id that will perform the docking maneuver with the vehicle currently selected. For example, if this maneuver was added to the AUV(i.e xplore-1) then the docking target would be the ASV (i.e. Caravela) and vice-versa.

**Number of Retries** - this is the way to the operator can retry the docking maneuver, if it's necessary. This means that the operator can choose up to a maximum of ten retries of docking attempt or if it is not pretended to retry the docking maneuver, he can choose zero retries.

**Number of Retries** - this parameter indicates to the low level tasks, what will be the function of the vehicle when performing the docking maneuver. With this, it is possible to have an unique docking maneuver task running in DUNE running simultaneously on both vehicles performing docking, doing independent tasks. For example, if its intended that the ASV(i.e. Caravela) performs the docking maneuver then the parameter in the *vehicle function* for it would be *station* while for the AUV, it would be the *target*.

## 5.3 DUNE Unified Navigation Environment approach

### 5.3.1 Introduction

This section intents solve the problems presented in *Chapter 4 - Problem Statement*. Has stated in previous chapter, DUNE is the onboard software that manages the interaction between the many parts of each vehicle. Some of the problems described on the previous chapter will be implemented using DUNE software.

### 5.3.2 Problem 2 - Docking Supervisor (ASV)

Given that the relationship between the two vehicles can be considered has master-slave one, one must consider which one is the master and other, the slave. In this case and given that the vehicle with the function "Station", i.e ASV *Caravela*, is the one choose to dock to the AUV then it will be considered the master. Then, the docking supervisor will be applied to the *Caravela*. This supervisor, which is essentially an function in *DUNE* (2.4.3) will be the one to control the phases of the docking maneuver, as stated in the docking sequence (5.2.1.2). It will also define the approach for the homing sequence taking into account the relative position between the two vehicles.

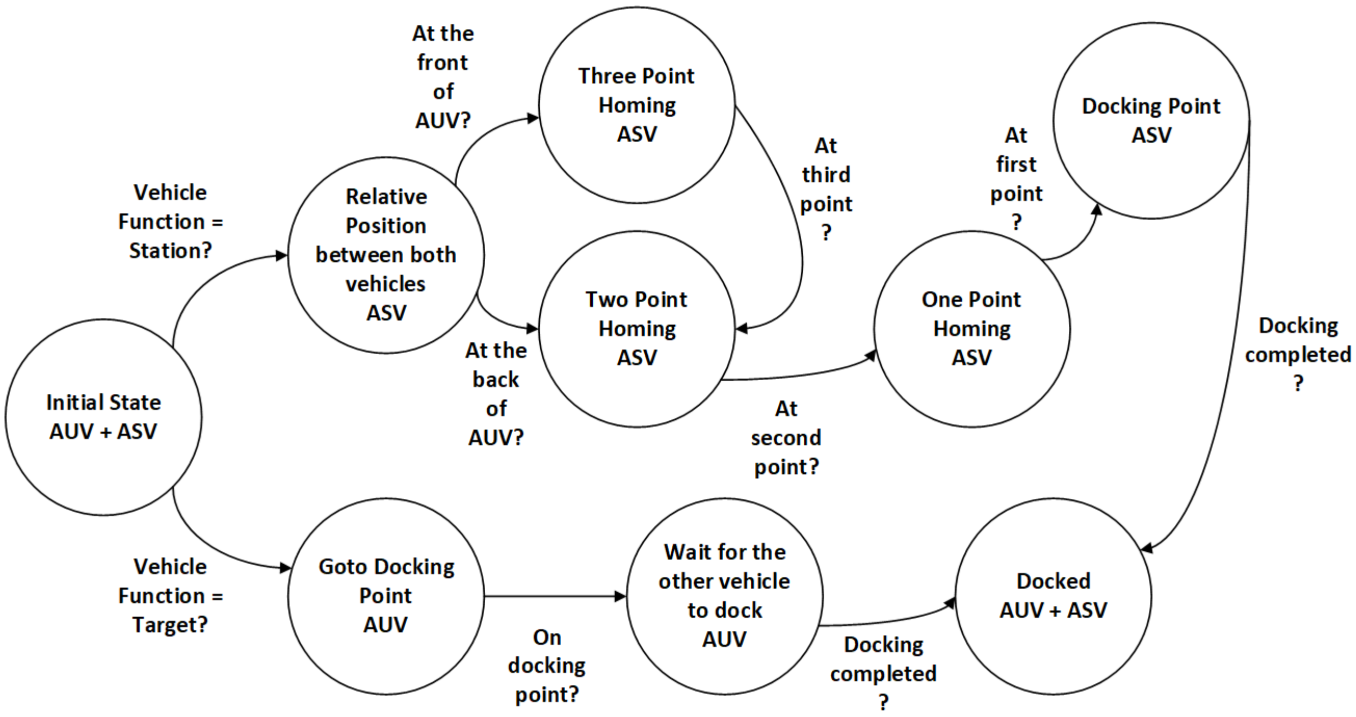


Figure 5.3: Docking Supervisor Diagram.

### 5.3.3 Problem 3 - Vehicle Position Correction Maneuver with External Disturbances (ASV and AUV)

Has stated in the previous chapter, when the vehicle needs to stay in a given position, be it for receive docking or wait for the other vehicle to get in position, it is needed to have an approach that solve that problem. This can be classified as a maneuver within a maneuver, given that it will be performed when the vehicle is performing the docking maneuver.

This maneuver will have two states described has follows:

1. **Inside SK radius** - when the vehicle is inside the acceptable radius based on the distance between the vehicle and the SK point, it will perform no action. Figure 5.4 shows the vehicle inside the radius performing no action despite being drawn be ocean currents.

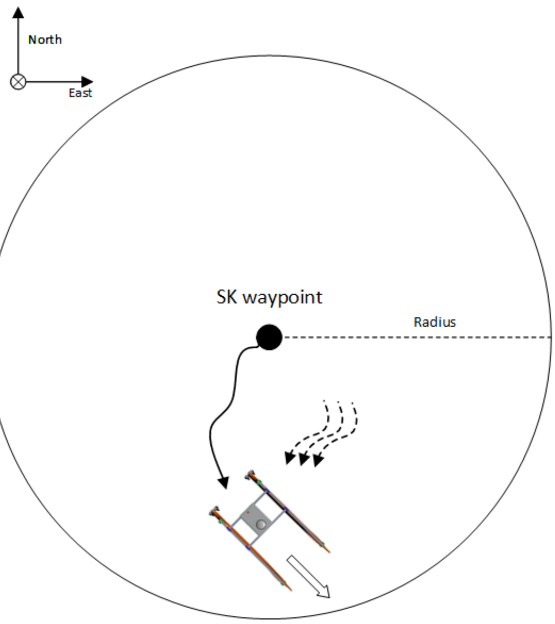


Figure 5.4: Caravela inside SK radius.

2. **Outside SK radius** - when the vehicle gets out of the defined radius, it will try to perform the position correction, moving to the center point of the SK circle. After being inside, the vehicle will stop any actions. Figure 5.5 shows the position correction of *Caravela*.

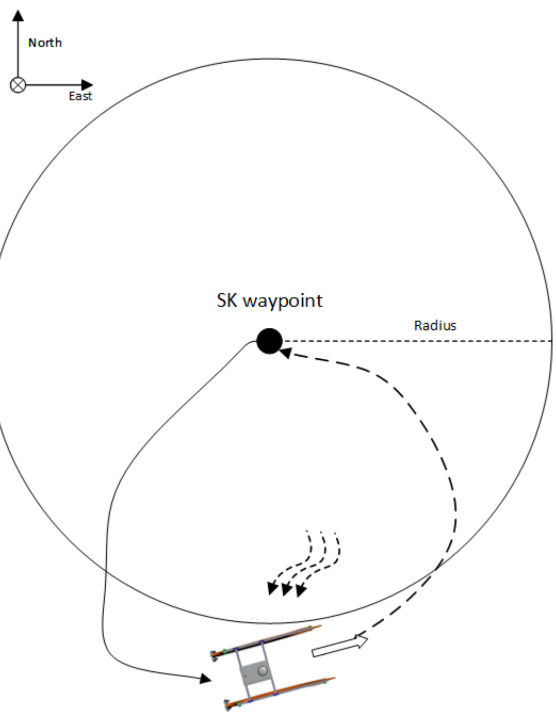


Figure 5.5: Caravela outside SK radius and correcting position.

To perform this maneuver, the vehicle must follow this sequence:

- A command to execute the SK maneuver is given to the vehicle which proceeds to the position designated.
- To know if it arrived to the position, it will calculate the distance between its actual position and the point which should perform the SK maneuver.
- After arrived to the point, it will keep calculation the distance between its position and the point and, when it is greater then the value of the radius of the SK maneuver, defined by the operator, it will move again to the initial position of the maneuver, returning to the SK area.

### 5.3.4 Problem 4 - Move to waypoint Maneuver (ASV and AUV)

Given a waypoint, it is needed to go from one point to another. This will be widely used in the docking maneuver because it is needed to the vehicles to move from one place to another. It will be used too to correct the heading of the vehicles without interfering with the low level controller. A simple example how the Goto maneuver works is given in figure

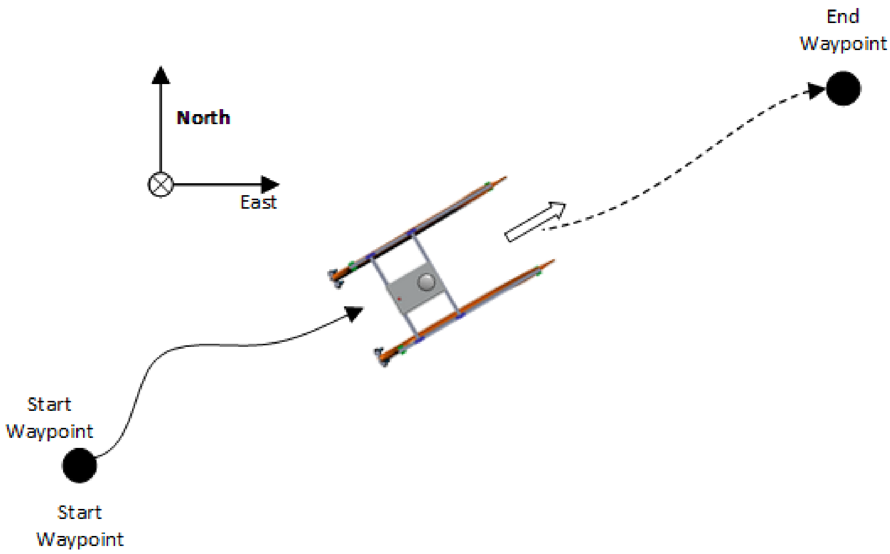


Figure 5.6: Goto maneuver from one point to another.

To perform this, a point is given by the supervisor, being it to perform the SK maneuver or to move the vehicle to a needed position or even to correct the heading of both vehicles.

### 5.3.5 Problem 5 - Homing of the vehicle (ASV and AUV)

This is an important problem to be solve since it will be the one that will place *Caravela* and a proper position relative to the AUV. When the AUV is in position and ready to perform the docking, *Caravela* will move to a point that makes it easier to perform the docking. This point will be determined by the relative position between both vehicles. For that it is necessary to verify the following conditions:

- *Caravela* is in front or in the back of the AUV.
- *Caravela* is on the left or the right of the AUV.

In figure 5.7 it is possible to see the different poses that *Caravela* can be and what would be the next move in order to perform the docking maneuver.

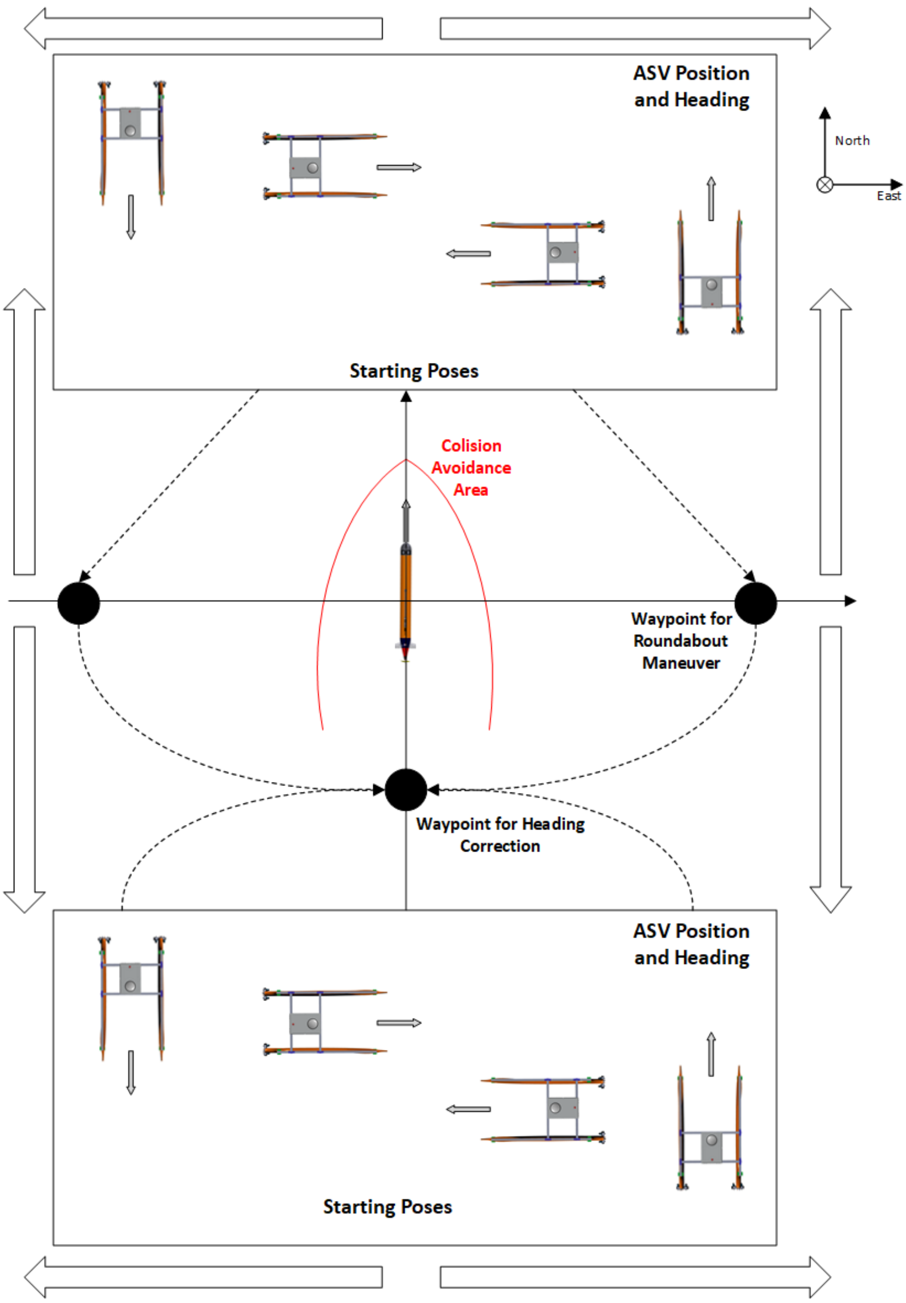


Figure 5.7: Multiple poses of *Caravela* relative to the AUV and the maneuver outcome.

1

Having the AUV as a reference point, there is two situations that needed to be considered. When *Caravela* is in front of the AUV with a given pose, it is necessary to perform a Roundabout maneuver. This maneuver will perform a circle around the AUV to avoid collisions between the vehicles and reach a position behind him.

#### 5.3.5.1 Roundabout Maneuver

In order to perform the docking maneuver without have any collisions between the two vehicles it is necessary to have some techniques that can prevent that. Collision Avoidance techniques are studied mainly in Automated Highway Systems (AHS) (Carbaugh *et. al. [38]*), in Air Traffic Management systems (ATM) (Tom *et. al.* [39]) and robotic manipulators (Schiavi *et. al.* [40]).

#### 5.3.5.2 Collision Avoidance System (CAS)

This system is required when the vehicle has to roundabout the AUV in order to be able to dock. An example can be seen in figure 5.8. Given that one of the assumptions described previously, that the ASV knows the state of the AUV and also taking into account the external disturbances, the following approach will not be about sensing techniques to detect the presence of obstacles but to define a maneuver that will enable *Caravela* to avoid the AUV.

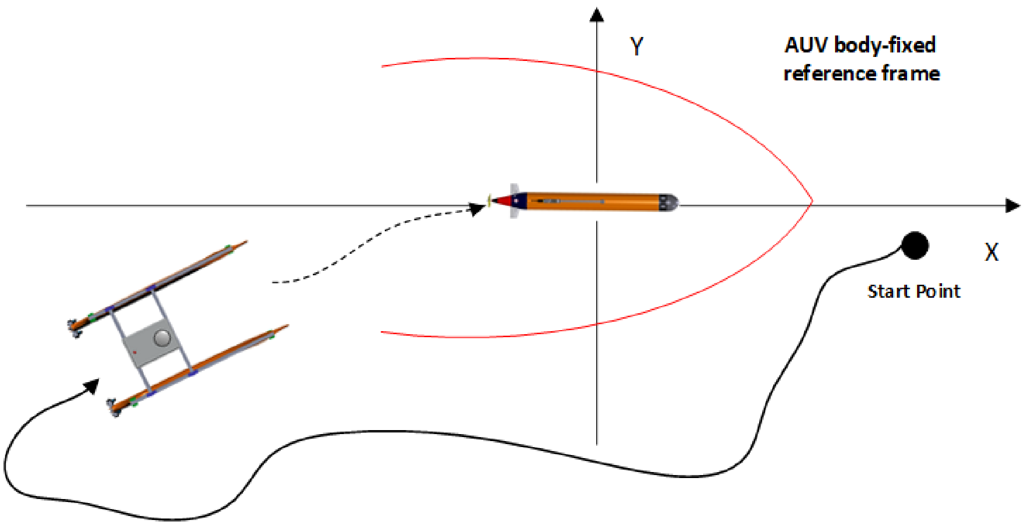


Figure 5.8: Roundabout maneuver to avoid collision.

To perform the docking and preventing the vehicles from collisions, two approaches where developed based on the previous situations:

- Three Point Homing.
- Two Point Homing.

In order to know which approach should be used at a given situation of the two vehicles, it is need to know the relative position between them. For that, there are three condition that needed to be defined in order to properly choose the right on. Have *AUV* as a reference vehicle, the condition are as follows:

- *Caravela* is in front of the AUV, on the right or on the left.

- *Caravela* in of the back of the AUV.

To know the relative position between *Caravela* and the AUV, in this case, if it is in the front or the back of the AUV, it is necessary to know the latitude of each vehicle which is the angle that is measured from the North Pole with a value of $+90°$, $0°$ in the equator and $-90°$ on the South Pole. Having the latitude of *Caravela* as a reference, and knowing that the values of latitude will only vary between $-90°$ and $0°$ on the northern hemisphere, it is necessary to verify that, when the latitude value of *Caravela* is higher than the AUV, it means that it will be further north than the AUV. However, this is not enough to know if the AUV is in front or in the back of *Caravela*. To know further, the heading of the AUV must be taken into account in order to know, if it is from 0 to $\pi$ and *Caravela* is at the front of the AUV, then it must contour it, which means it must perform the *Three Point Homing* described later. Otherwise, *Caravela* is on the back of the AUV and only has to perform the *Two Point Homing*. When the AUV has a higher value of latitude, the approach reverses, which means, when the heading of the AUV is from 0 to $-\pi$, *Caravela* must contour it.

When *Caravela* is in front to the AUV, it is necessary to know if it is on the right or the left of the AUV, to perform the roundabout (Three Point Homing) with the least path. To do this it is necessary to calculate the heading difference of both vehicles and defining that when the difference is greater than zero, *Caravela* will be on the right of the AUV and when it is lesser than zero, it will be on the left of the AUV.

#### 5.3.5.3 Three Point and Two Point Homing

It is used when *Caravela* need to go around the AUV, to prevent collision between the two vehicles. It has the following sequence:

**Three Point Homing** - If *Caravela* is in front of the AUV, it will go to a point defined sideways from the AUV as seen in figure 5.9.

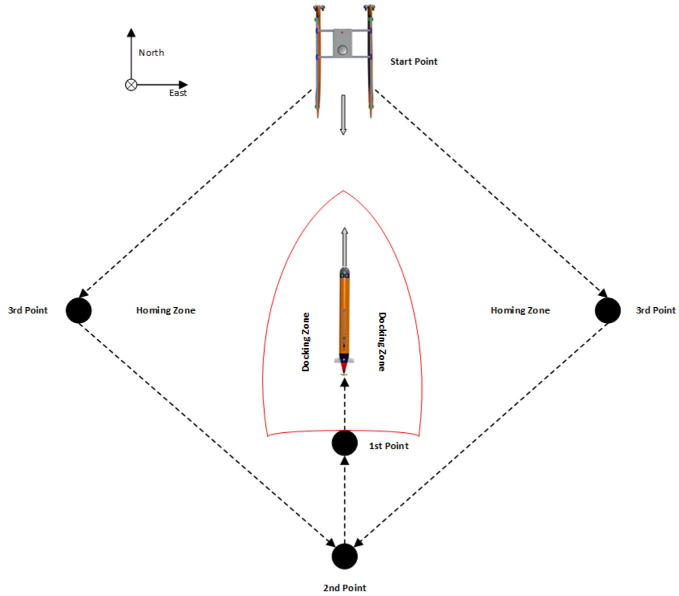


Figure 5.9: Third Point Homing Sequence

It will find if it is on the left or the right of the vehicle and will choose the point that gives *Caravela* the least distance path. This third point is defined within 50 meters from the AUV and it is pointed on the opposing direction of the heading of the AUV. This means that *Caravela* can reach this point safely without risking a collision. Figure 5.10 shows the point being defined by the supervisor, for *Caravela*

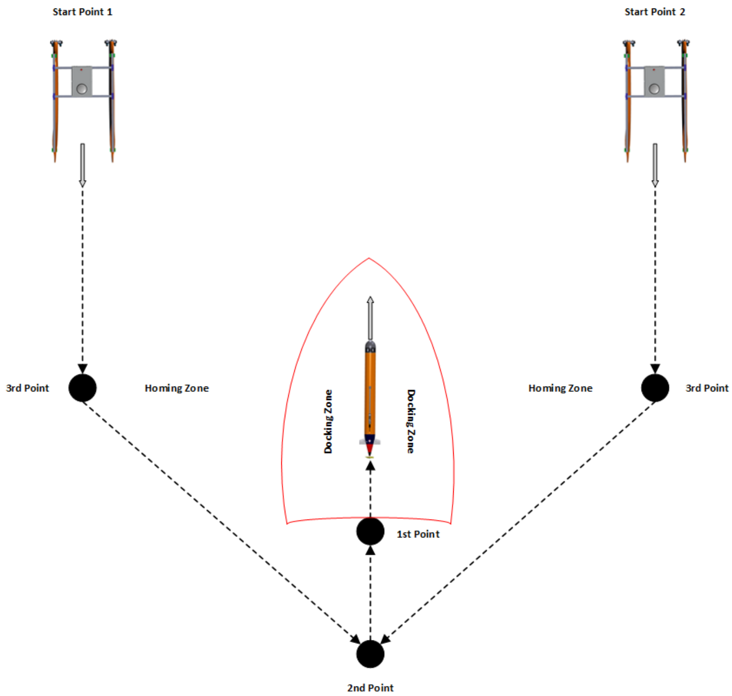


Figure 5.10: Third point in the Homing Sequence

**Two Point Homing** - If *Caravela* is in the back, whether is coming from the third point sequence or is starting the homing sequence now, a point 50 meters far from the AUV will be defined. This second point in the homing will place *Caravela* in a proper position but not with the right heading to perform the docking sequence. Figure 5.11 will show **Caravela** heading to the second point after passive the third point.

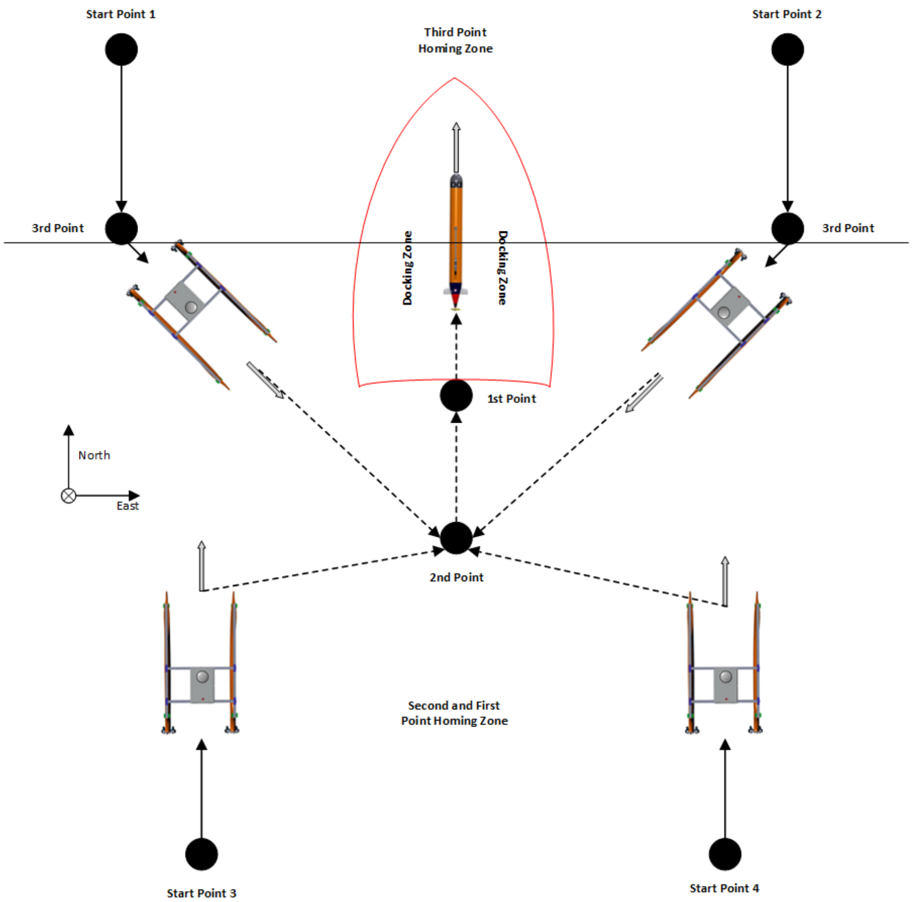


Figure 5.11: Second point in the Homing Sequence

After reaching the second point, *Caravela* should proceed to the first point, which will be used to correct is heading, placing it with the same value has the AUV. This point will be defined with half the value of distance from the previous on, for example if the second point is placed at 50 meters from the AUV, the first point will be 25 meters from the vehicle. This ensures that **Caravela** have enough room to perform the maneuver and get the same heading of AUV when reaching the point. Figure 5.12 shows *Caravela* reaching the defined point.

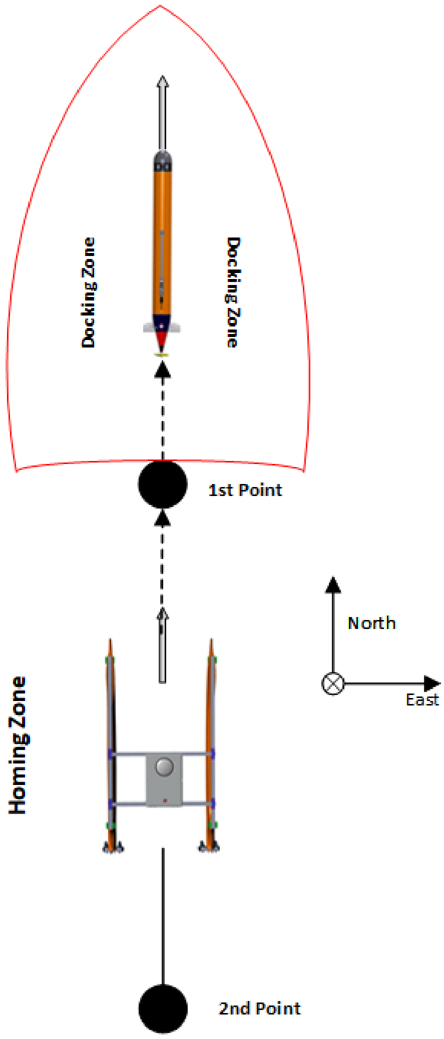


Figure 5.12: First point in the Homing Sequence

### 5.3.6 Problem 6 - Docking of the vehicles (ASV)

After execution the homing sequence and the vehicle have the right heading, the final approach of the maneuver, docking, must be done. This will define a point right at the position of the AUV, with a given error, which *Caravela* will try to reach. It will still calculate the distance between the ASV and the AUV, lowering the speed in proportion of the decreasing distance. This means when *Caravela* is reaching the AUV, its speed will be close to zero and it will have actuating in it the dragging forces. Figure 5.13 shows *Caravela* point to the position of the AUV after finished the homing sequence.

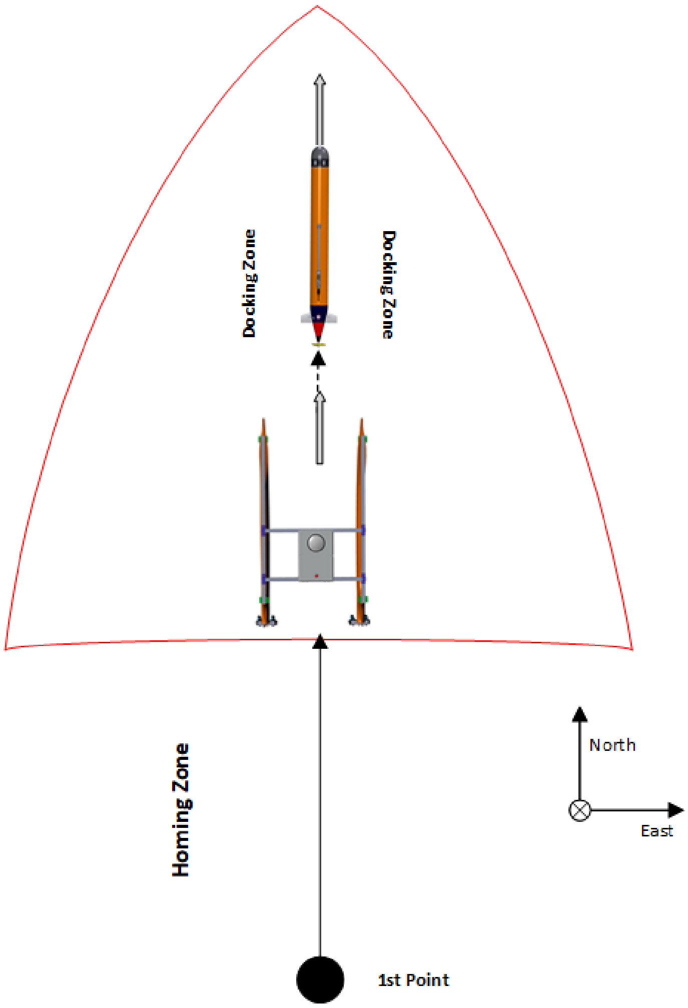


Figure 5.13: Docking Sequence

### 5.3.7 Problem 7 - Abort sequence (ASV and AUV)

This sequence will be present in both vehicles and will be considered in the following situations:

1. *Caravela* is performing the homing or docking sequence and AUV need to correct its position (SK maneuver).

2. Both vehicles have the same function (Target or Station)

3. *Caravela* detects that it is not possible to perform docking when reaching the AUV in the final stages of the docking sequence.

4. The AUV or *Caravela* timeouts when one of the vehicles is performing a task and the other is waiting.

On the first *Caravela* will stop its maneuvering and will wait for the AUV to reach the SK point and start all over again.

On the second situation, when both vehicle start the docking maneuver, they will detect if any of the vehicles on the bus have the same function has theirs and will abort the maneuver and ask for the operator to give unique function to each vehicle.

On the third situation, *Caravela* can detect if there is the possibility to perform successfully the docking maneuver based on its path and the error associated with the position to be reached. If it detects and the operator place an order to retry the docking maneuver, it will start again the homing sequence and then the docking sequence. If not, the vehicle will stay on its actual position waiting for new orders.

The forth situation was made thinking that, when there is an interaction between two independent systems, a deadlock can happen. This means that we can have vehicles in an infinite loop and cannot perform the docking maneuver. For that it was implemented a timer based system on the vehicle. This means that it will be used when a vehicle is waiting for other while the former is performing some task. For example, when *Caravela* is waiting for the AUV to reach the docking waypoint, it will activate the timer with a given value and wait for the AUV to finish the task. If it does not finish the task until before the end of the timer, *Caravela* will abort the maneuver, preventing a perpetual deadlock. This can be used in other situations like when the AUV is waiting for the *Caravela* to finish the *Homing* and the *Docking* of the docking maneuver.

### 5.3.8 Implementation

As stated before, the implementation of the approach was made using DUNE software, running on the vehicle. To test all the implementation, one would run Neptus in its normal form and would run simulation or both vehicles that would be recognized by Neptus as actual vehicles. To test in a situation closer to the reality, it was added currents by placing speed values that would force the vehicle to North or East.

The Docking maneuver have the supervisor that oversee the all the inputs and outputs and act accordingly. This supervisor manages the state machine used to know the approach for the homing point, which point to choose given the relative position of the vehicles, the docking approach, will send and receive messages to synchronize the docking maneuver between the two vehicles, it give different function for the vehicles given it different function, assigned in Neptus. It have a docking timeout that will help the vehicles to overcome a deadlock that could appear on the state machine of the docking maneuver, like one vehicle waiting for the other to finish something that it is not doing.

The maneuver have some sub-maneuvers that exist in DUNE but had to be implemented inside the maneuver, given the nature how the maneuver section of DUNE was built.

The synchrony between the two vehicles would be made with the message *DockingState* given that this message would be sent from one vehicle to the other using an UDP socket.

#### 5.3.8.1 DockingUSBL task

There is a task implement that would mimic an USBL modem given that it would take the encapsulated IMC message *DockingUSBL* with an *EstimatedState* within and would send out to the docking supervisor a range through the IMC message *UamRxRange* and angle with the message *USBLAnglesExtended* as a normal USBL modem would do and calculate the relative position between the two vehicles.

## 5.4 Conclusions

In this chapter we presented the approach to the docking maneuver given the systems available.

It was presented the docking sequence of the maneuver with each individual phases (5.1).

The toolchain approach was described (5.2), where it was shown the three IMC messages for the docking maneuver, named *Docking*, *DockingState* and *DockingUSBL* (5.2.1).

Neptus approach for the interface between the vehicles and the operator and how it will work with the IMC messages generated (5.2.2)

The description of each problem that should be solved with DUNE (5.3) is made in this section, where there is a brief explanation of the station keeping maneuver (5.3.3) and go to waypoint maneuver (5.3.4). It was presented the homing approach for the docking maneuver (5.3.5), the docking approach (5.3.6) and the abort approach (5.3.7).

# Chapter 6

# Results

This chapter presents the results obtained from the application of the approach from the Chapter 5 - *Approach.*

## 6.1 Test Plan

This section presents the test plan built to test the approach of the docking algorithm explained in the previous chapter. Obviously, the plan does not test every possibility, however it shows the basic approaches to the docking problem.

The test for Problem 1 intents to show the maneuver interface added to Neptus software.

The test for Problem 3 and 4 shows the development of the Goto Waypoint and Station Keeping maneuver.

The test for Problem 2, embedded in Problem 5 and Problem 6 shows the docking supervisor in action, to choose the best approach for the docking based on the relative position between the two vehicles.

Problem 5 and 6 will be displayed in four scenarios, to test the collision avoidance technique implemented. Hence,

- Case 1 - ASV is on front of the AUV
- Case 2 - ASV is on front and right of the AUV
- Case 3 - ASV is on front and left of the AUV
- Case 4 - ASV is on back of the AUV

To test Problem 7, a series of conditions will be applied in order to the vehicles abort their operations.

## 6.2 Problem 1 - Docking Maneuver Controller

This section shows the approach to the interaction between the operator and the vehicles to perform the docking maneuver, using the Neptus interface. Figure 6.1 shows the initial screen where it is possible to make or send plans to the vehicles, see them in real-time and with real positions.

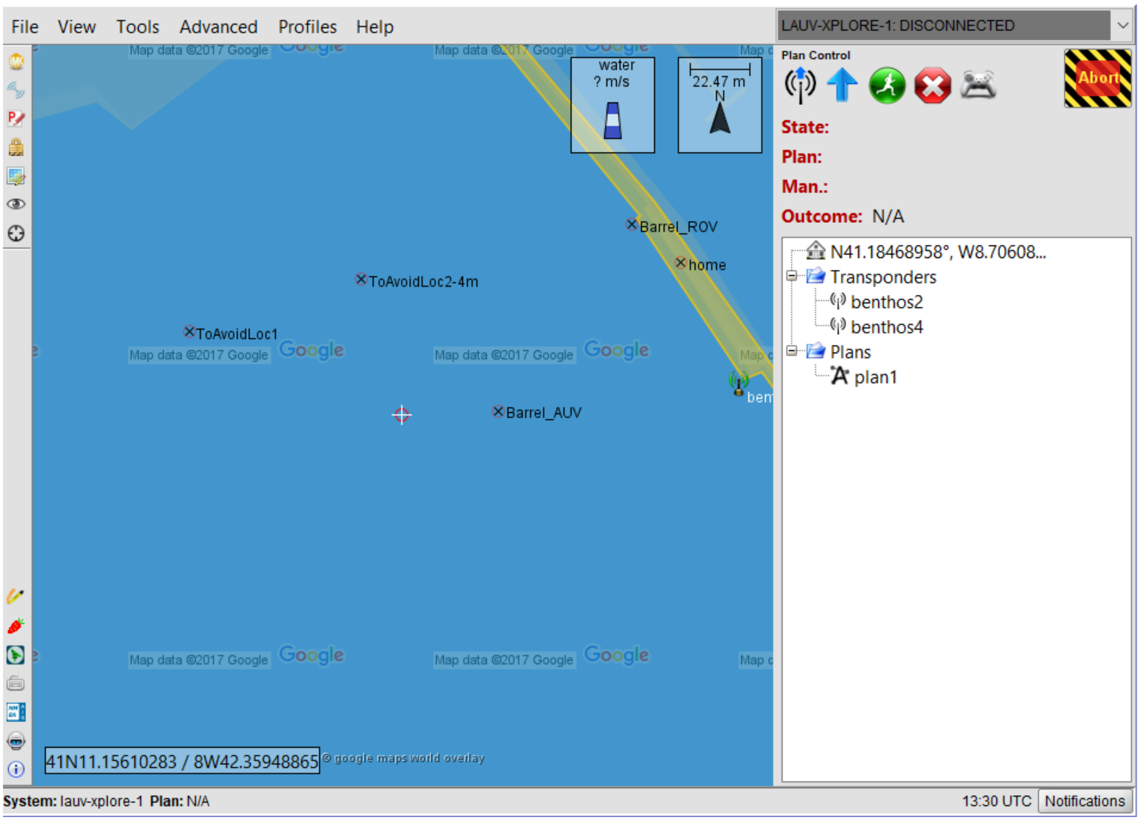


Figure 6.1: Neptus AUV console

To produce a plan it is necessary to open a menu on the left side. This will enable the vehicle which the plan will be made and choose what type of maneuver is pretended. This can be seen in figure 6.2

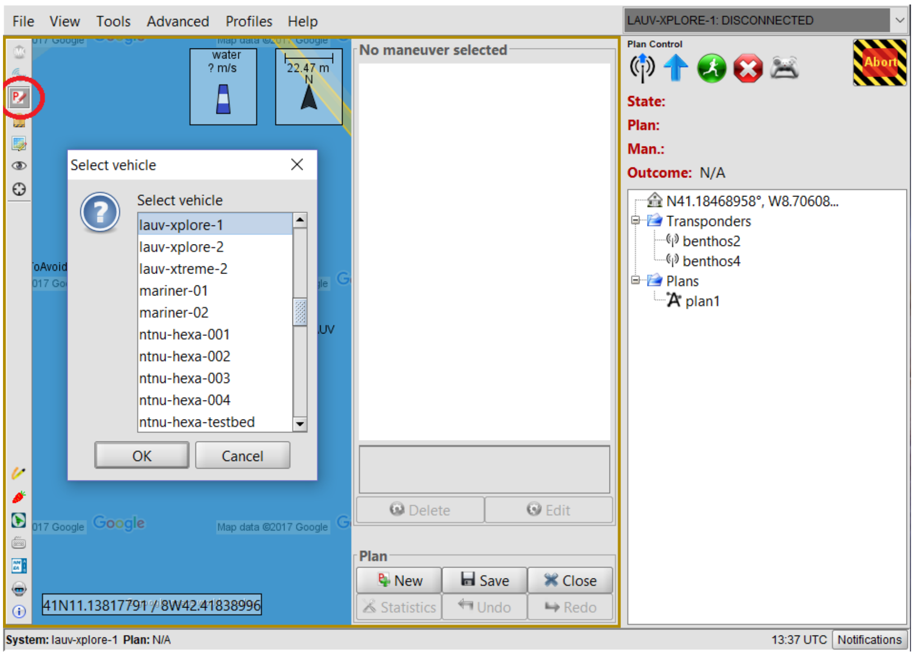


Figure 6.2: Neptus plan editing

After selecting the vehicle, it is need to choose the docking maneuver as seen in figure 6.3.

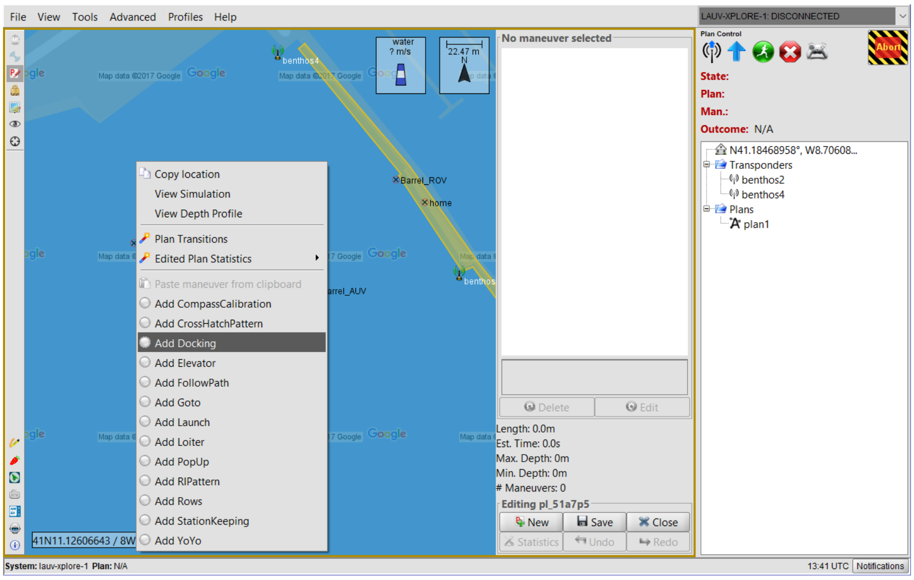


Figure 6.3: Neptus docking maneuver selection

When adding the maneuver, as stated in the previous chapter, it is need to add some parameters in order to the vehicle perform the docking. In this case, given the vehicle chosen for the example is *xplore-1* the parameters to perform with the ASV *caravela* should be:

- **Docking Target** - caravela

- **Number of Docking Retries** - 5
- **Vehicle Function** - Target

This can be seen in figure 6.4.

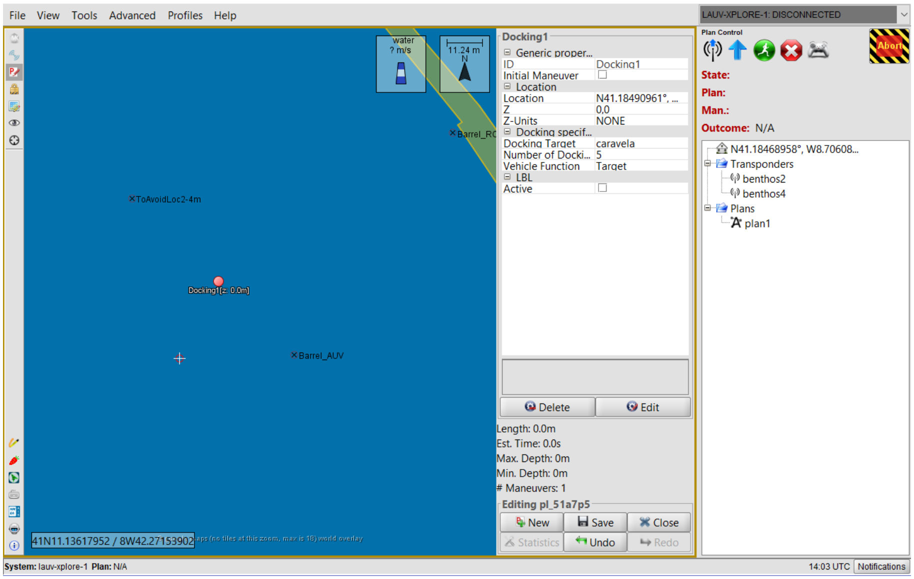


Figure 6.4: Neptus docking parameters completion

After this, the operator only need to save the plan or add other maneuver, if intended and send it to the vehicle. This plan needs to be generated to each vehicle in the docking maneuver.

## 6.3 Problem 3 - StationKeeping Maneuver and Problem 4 - Goto waypoint Maneuver

This section intents to show the implementation of two side maneuvers that are need to perform the docking maneuver. This maneuvers are implemented solo in Neptus, although one added a similar approach for both of them inside the docking maneuver.

### 6.3.1 Problem 3 - StationKeeping Maneuver Implementation

Considering that it is need to keep the vehicles in a given position, it was implemented a maneuver that would do that.

The vehicle would fix its position and when outside the maximum distance defined, it would correct its position and return to the initial pose. For that, it is needed to know the distance between the actual pose of the vehicle and the given point of station. Thus, it is calculated the distance between the two points.

A displace of coordinates is done for the actual position of the vehicle using the latitude and longitude coordinates of the previous fix and added the values of x and y. This will output the actual latitude and longitude of the vehicle. After that, it is calculated the distance between the two poses, converting each latitude and longitude to ECEF and adding each x and y that result from that. The square root of all the values will be the distance.

The result of this implementation can be seen in figure 6.5 where it is shown the ASV *Caravela* inside the defined area.

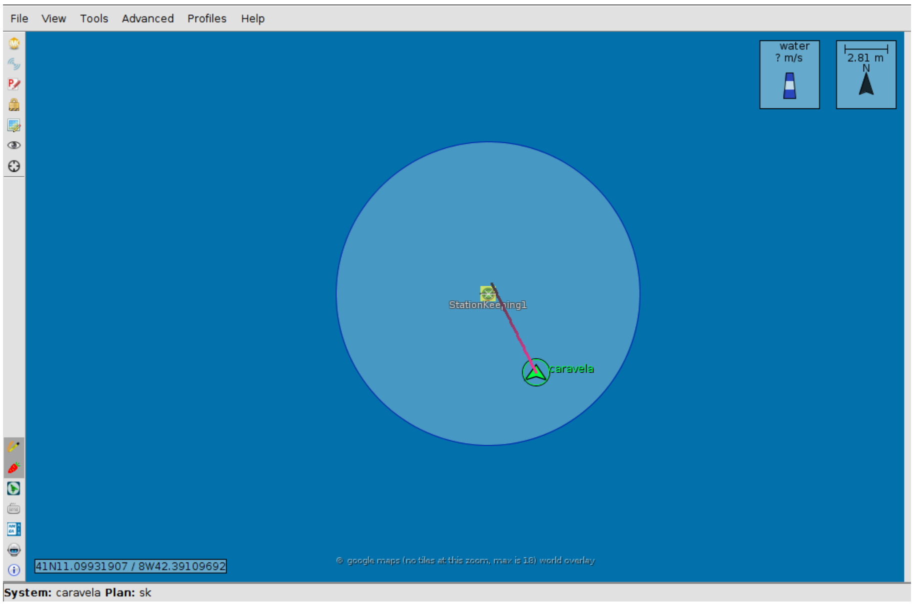


Figure 6.5: Inside Station Keeping

When external disturbances put the ASV outside the defined area, it will correct its position, as seen in figure 6.6.

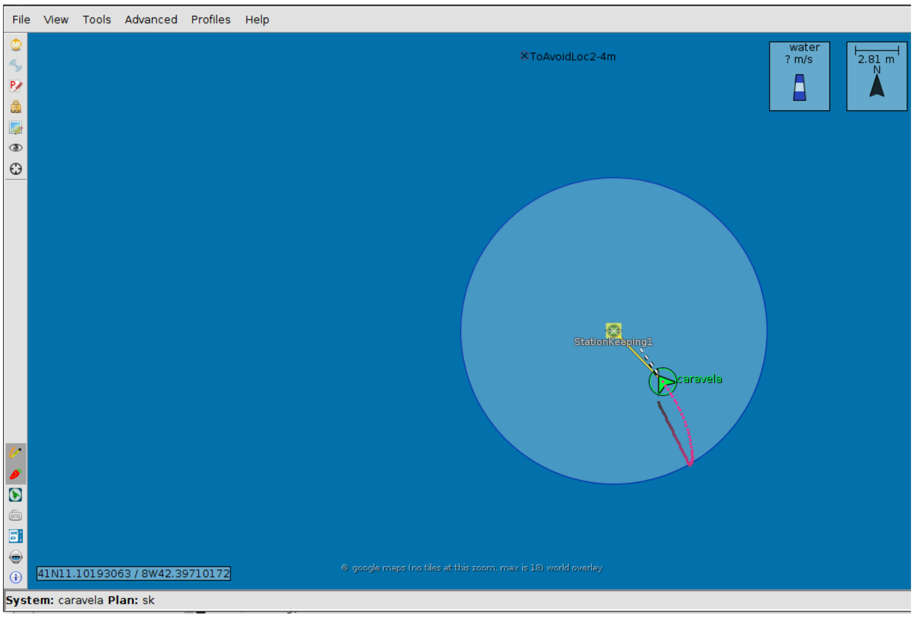


Figure 6.6: Outside Station Keeping

### 6.3.2 Problem 4 - Goto waypoint Maneuver Implementation

To move the vehicles to a given waypoint, it is necessary to add a submaneuver that moves the vehicle to a pretended destination. For this, it is only necessary to fill a IMC message named *DesiredPath* with the latitude and longitude desired. This message will be sent to the IMC bus a read by low level controller that will send the vehicle to the point. This submaneuver can be seen in figure

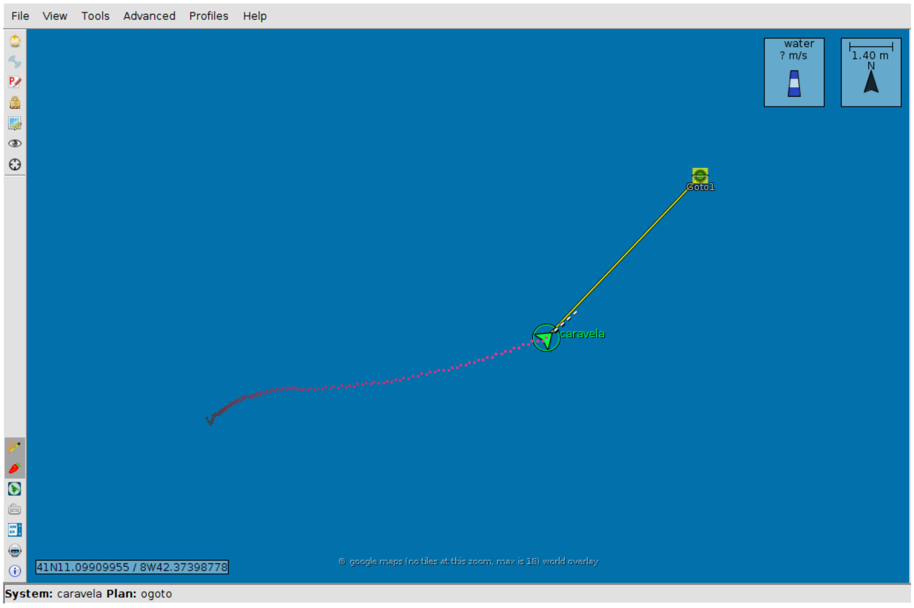


Figure 6.7: Goto submaneuver

With the implementation of this two submaneuver it is possible to solve the following problems.

## 6.4 Problem 2, Problem 5 and Problem 6

This section intents to solve the problems 5 - Homing and problem 6 - Docking having the problem 2 embedded with them.

### 6.4.1 Problem 2 - Docking Supervisor

This supervisor will be the core of the maneuver, since it will managed all the inputs and outputs. It will run on the ASV. This means that it will choose when to start the Homing or Docking Sequence, it will send and receive messages to the AUV and it will manage when the maneuver ended or should be aborted.

When the start trigger is sent from Neptus, with all the parameters referred previously, the supervisor will see the function of the actual vehicle and perform action accordingly. Given that the vehicle with the function *Station* is the one that will perform most of the docking maneuver, the supervisor will see the availability of both vehicles to perform the maneuver. If both are available, it then proceeds to know the relative position between them. This means that it will see if it is on the front or the back, right or left of the other vehicle. After knowing the relative position, it will choose the Homing sequence to start, Three Point or Two Point. After the sequence is finished it will start the Docking Sequence. The supervisor will too see if it is need to retry the maneuver, based on the information provided by Neptus. In any moment an abort event can be detected and the supervisor will stop the maneuver and react accordingly. It will too send messages and receive through an UDP socket, between the vehicles in order to have a synchronized maneuver.

### 6.4.2 Problem 5 - Homing and Problem 6 - Docking

As stated before, a plan of tests was made with five cases, in order to test most of the situations that can occur when performing the docking maneuver.

#### 6.4.2.1 Case 1 - ASV is on front of the AUV

Given that it is need to avoid collisions between the vehicle and for that, perform a roundabout around the AUV, it was tested when the ASV is in front of the AUV in a relative manner. In this case, the ASV performed a Goto to a given point, to place it in front of the AUV, and then perform the docking. Taking into account that it is virtually impossible to be precisely at the front of the AUV, the ASV will always choose to go to the right or the left of the AUV. In this case, it is needed to perform the Three Point Homing.

In figure 6.8 it is possible to see both vehicle performing the test case.

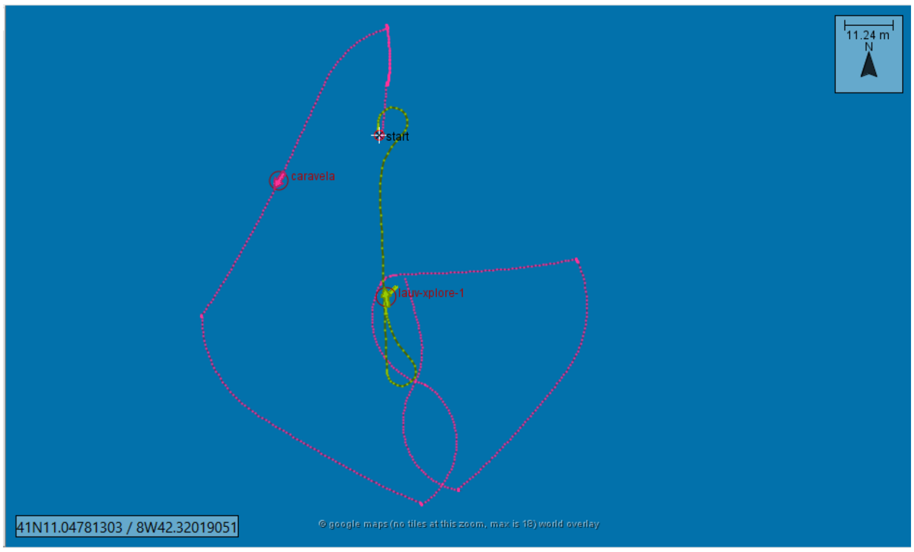


Figure 6.8: Case 1 - ASV and AUV performing the docking

Figure 6.9 shows the sequence of the vehicles.

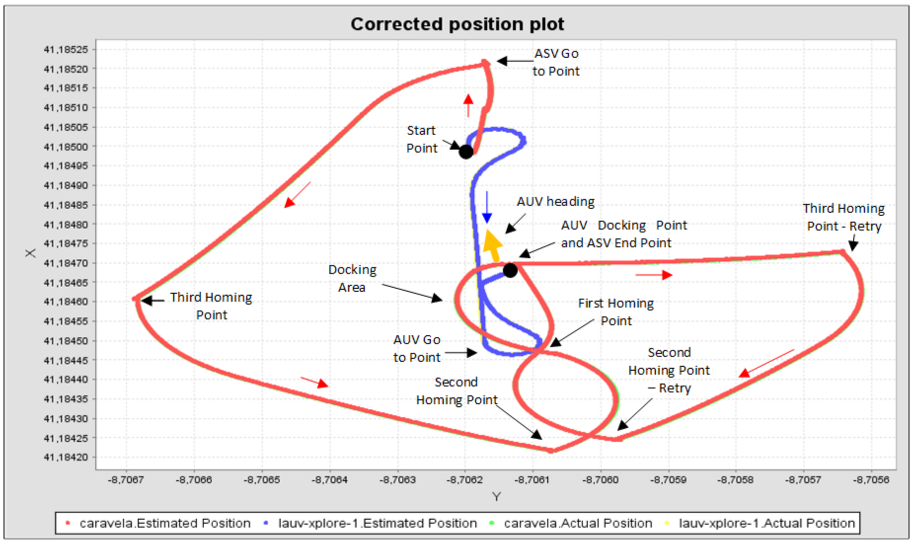


Figure 6.9: Case 1 - ASV is on front of the AUV vehicle paths

Both vehicles start at the same point, performing first, Goto waypoints in order to place them with the conditions necessary to test this case. The AUV will go to a point behind the initial point, and then will move back to have it's heading point North. When the AUV is going to the docking point, the ASV waits in StationKeeping for the AUV to finish its maneuver. While the ASV waits, it is dragged by the ocean currents, added in the simulation. After the AUV reaches the docking point, it send a message to the ASV, indicating the availability to perform the docking. Since the ASV in at the front and, in this case at the left of the AUV when the docking maneuver is starting, it will perform the Three Point Homing Sequence on the left of the AUV. This point will be defined with a distance from it of 50m to ensure space to the ASV to maneuver. After reaching the Third Homing Point, it will move to the Second Homing Point and after that the First Homing Point, before reaching the Docking area. After reaching it, it will start the Docking Sequence, which will only send the vehicle to the exact position of the AUV. In this case, it is possible to see that a docking retry was done, since the first attempt was failed because of the ASV not reaching the AUV. When it retries, it will start again the evaluation of the situation, meaning that it will again find the relative position between the two vehicles. In this case, since in the it considered that the ASV was on the right and front of the AUV, it placed the Third Homing Point on the right and repeated the sequence described before. In this case, it was possible to make the docking of the two vehicles but only on the second try.

#### 6.4.2.2 Case 2 - ASV is on front and right of the AUV

Similar to the previous case, here too the ASV must avoid collisions. Figure 6.10 shows an example of the test in Neptus.

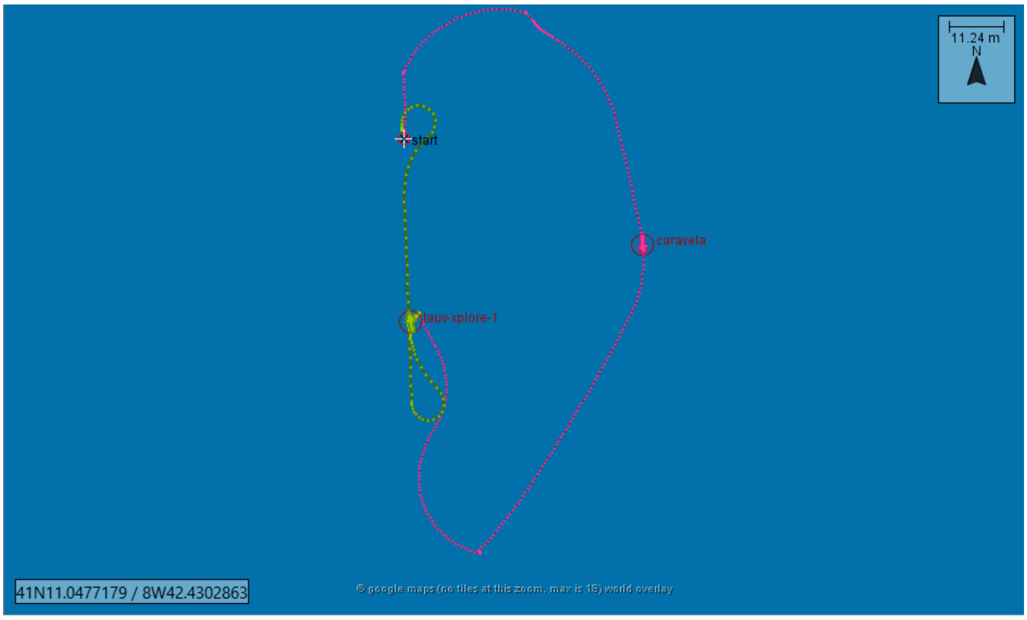


Figure 6.10: Case 2 - ASV and AUV performing the docking

As stated before, both vehicles will move to a determined point in order to perform the test case. Since in this case the ASV is still in front of the AUV it will start with the Three Point Sequence placing the Third Homing Point on the right of the AUV. After reaching the Third Homing Point it will proceed to the Second Homing Point and finally the First Homing Point. The difference from this case from the previous on is that it will start the Three Point Homing on the right of the AUV and it can perform the docking with only one attempt. This sequence can be seen in figure 6.11.

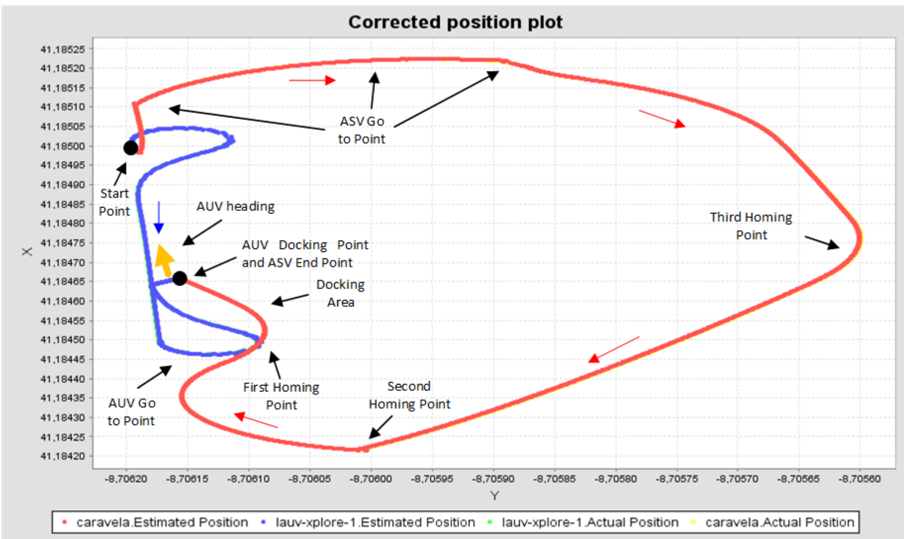


Figure 6.11: Case 2 - ASV is on front and right of the AUV vehicle paths

#### 6.4.2.3 Case 3 - ASV is on front and left of the AUV

This case is very similar to the previous cases since it happens when the ASV is in front of the AUV. Figure 6.12 shows the vehicle performing the docking maneuver using the Three Homing Sequence.

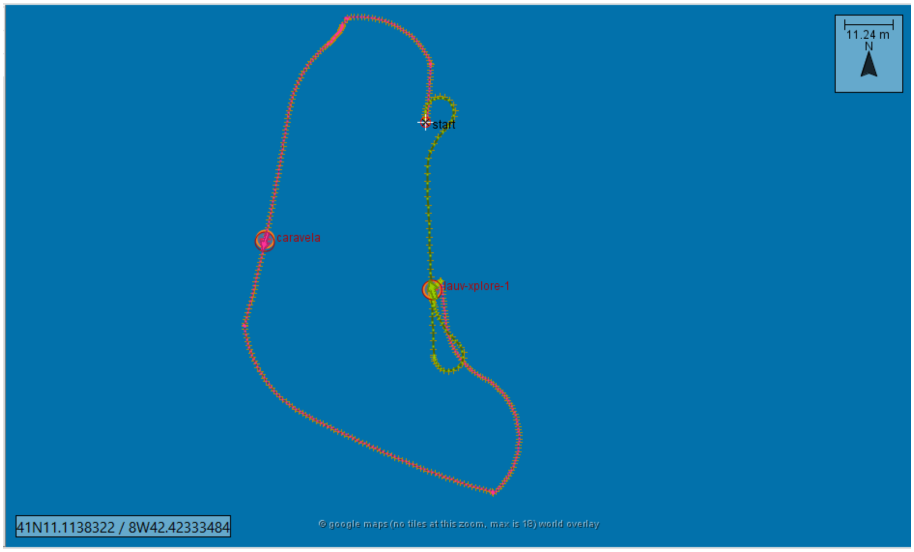


Figure 6.12: Case 3 - ASV and AUV performing the docking

The vehicles get in place to perform the maneuver and, in this case, the Third Homing Point is placed on the left of the AUV, has expected. It follows the same sequence as the cases before, and finishes the docking with success. The docking sequence can be seen in figure 6.13

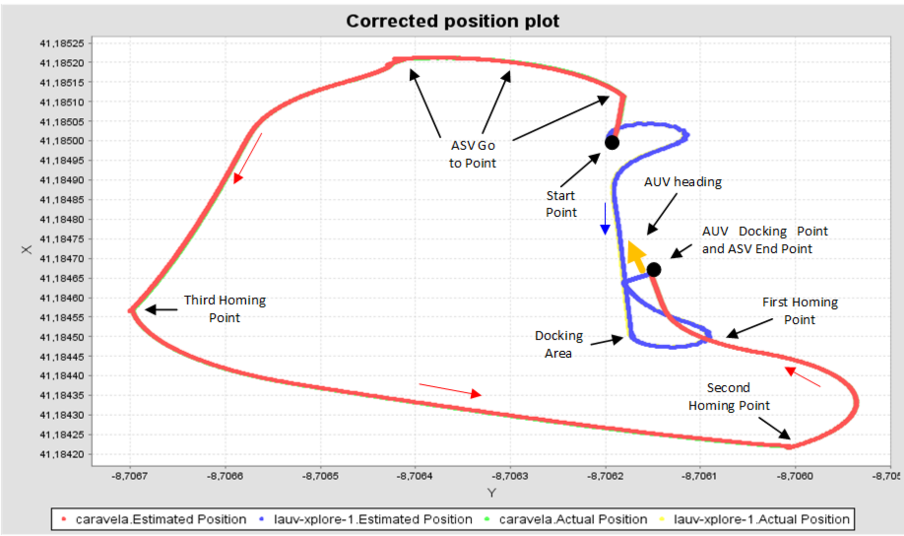


Figure 6.13: Case 3 - ASV is on front and left of the AUV vehicle paths

#### 6.4.2.4 Case 4 - ASV is on back of the AUV

This case shows a difference from the previous cases. Before, the ASV was always in front of the AUV but in this case, it will be on the back. This means that the Homing Sequence will change since it only need to perform the Two Point Homing. In this case, doesn't matter if the vehicle start the Homing Sequence on the right or the left of the AUV since the Second Homing Point will only rely on the heading of the AUV. In figure 6.14 and figure 6.15 it is possible to see two tests made when the ASV is on the right and on the left of the AUV, resulting in a similar Homing Sequence.


Figure 6.14: Case 4 - ASV and AUV performing the docking

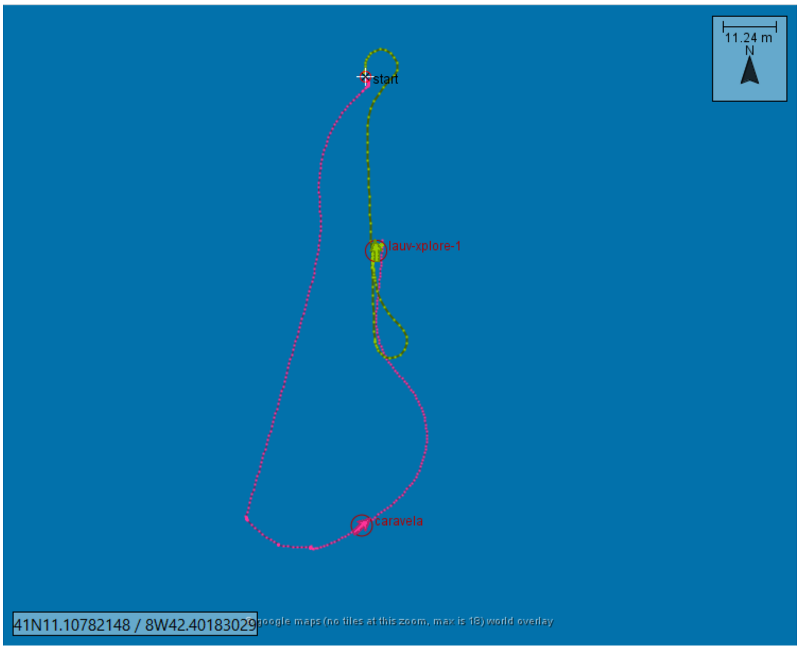


Figure 6.15: Case 4 - ASV and AUV performing the docking

With this is possible to analyze the sequence given that the ASV can start its maneuver from the left, right or center, the outcome of with will be always the same.

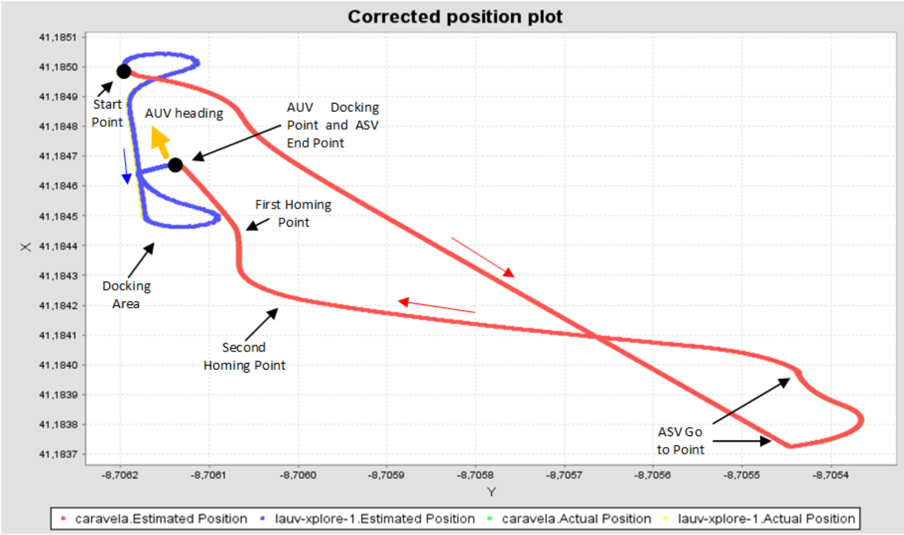


Figure 6.16: Case 4 - ASV is on back and right of the AUV vehicle paths

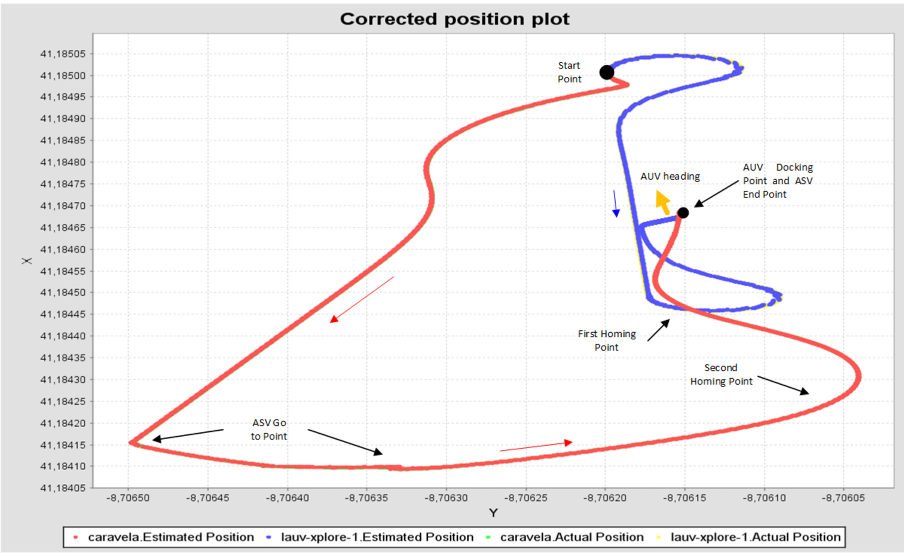


Figure 6.17: Case 4 - ASV is on back and left of the AUV vehicle paths

#### 6.4.2.5 Homing Test Conclusion

The homing sequence is one of the most important of the docking maneuver since it will enable both vehicles to perform the maneuver without the danger of a collision.

Table 6.1 shows a resume of the outcome of all the cases presented previously and what changes, given the relative position of both vehicles.

| Case | ASV position relative to AUV | Homing | Docking Success | Number of Tries |
|---|---|---|---|---|
| 1 | Front | Three Point Homing | Yes | 2 |
| 2 | Front and Right | Three Point Homing | Yes | 1 |
| 3 | Front and Left | Three Point Homing | Yes | 1 |
| 4 | Back, Right or Left | Two Point Homing | Yes | 1 (Each) |

Table 6.1: Resume of the test cases of the Homing Sequence

Although this tests doesn't show that the docking maneuver will always work at any given conditions, it at least represents the worst cases scenarios for the docking maneuver.

## 6.5 Problem 7 - Abort

As indicated on the previous chapter, Abort will have at least four conditions that needed to be considered:

1. AUV (ASV) aborts actual maneuver to wait for the ASV (AUV) to perform a Station Keeping.
2. Both vehicle have the same function.
3. Not possible to finalize docking maneuver, proceeds to abort and retry if needed.
4. Docking maneuver timeout to prevent deadlock.

For the first situation, when one of the vehicles is performing a Station Keeping as described in 6.3.1 the other should stop the actual task and wait for the first one to finish the correction. In order to have this kind of synchronization, the state of availability in the vehicle performing the SK is changed and the a message is sent to the other vehicle. With this, it is possible to prevent collisions of the vehicles. In figure 6.18 it is possible to see that the ASV stopped its maneuvering and is waiting for the AUV to correct its position.

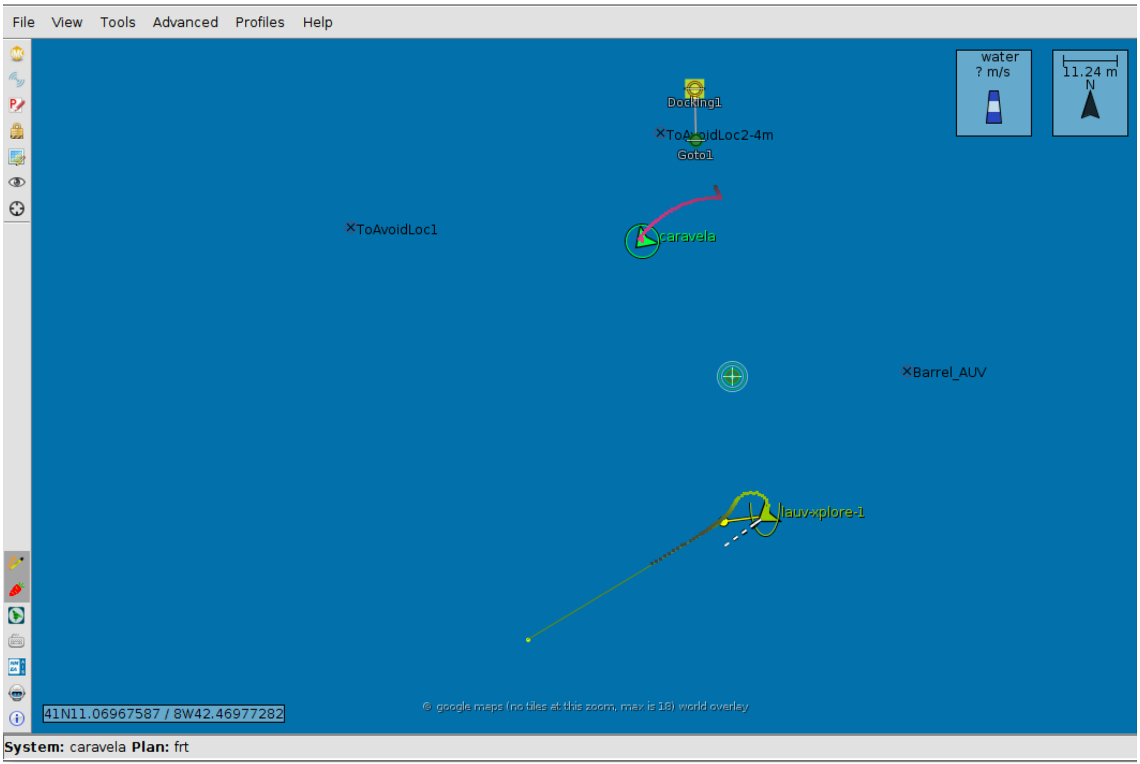


Figure 6.18: ASV abort and waiting for the AUV to finish SK

For the second situation, there shouldn't two vehicles with the same function performing the docking maneuver on the same time. When the vehicles start the maneuver, they will see on the

IMC message bus to a vehicle that have the same function has theirs. When this happens, both vehicles stop the docking maneuver and present the operator with a error message, has seen in figure 6.19.

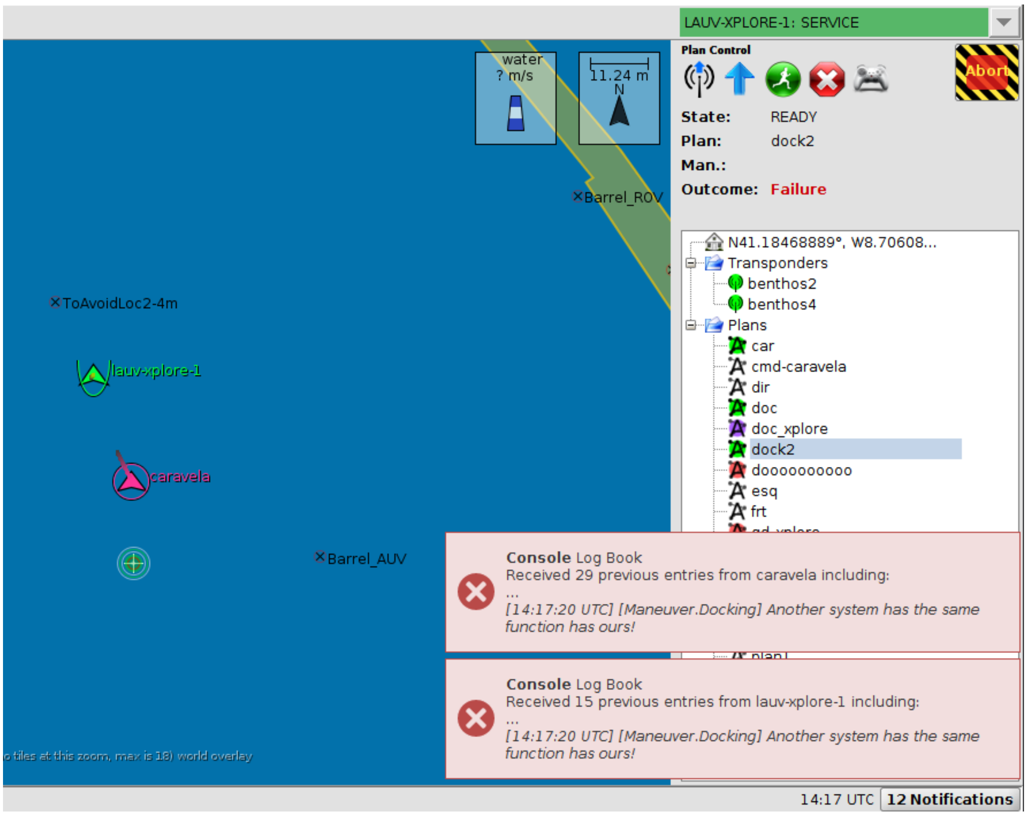


Figure 6.19: Output error in Neptus when two vehicles have the same docking function.

On the third situation, when the ASV tries to finalize the docking but fails to do it with success, there are two options that can be taken into account: one, both vehicles stop the docking maneuver and output to Neptus that was not possible to finish the maneuver or, two, if there is a number of retries, the ASV will try again to perform the docking. This can be seen in a figure presented previously, figure 6.9 where the ASV tries a second time to perform the docking maneuver.

For the fourth situation, a docking timeout was implement. This timer will start count when one of the vehicles is waiting to finish a task critical for the docking maneuver. When the timer reaches a value defined previously, the docking maneuver will stop and output an error associated with this abort. This output can be seen in figure

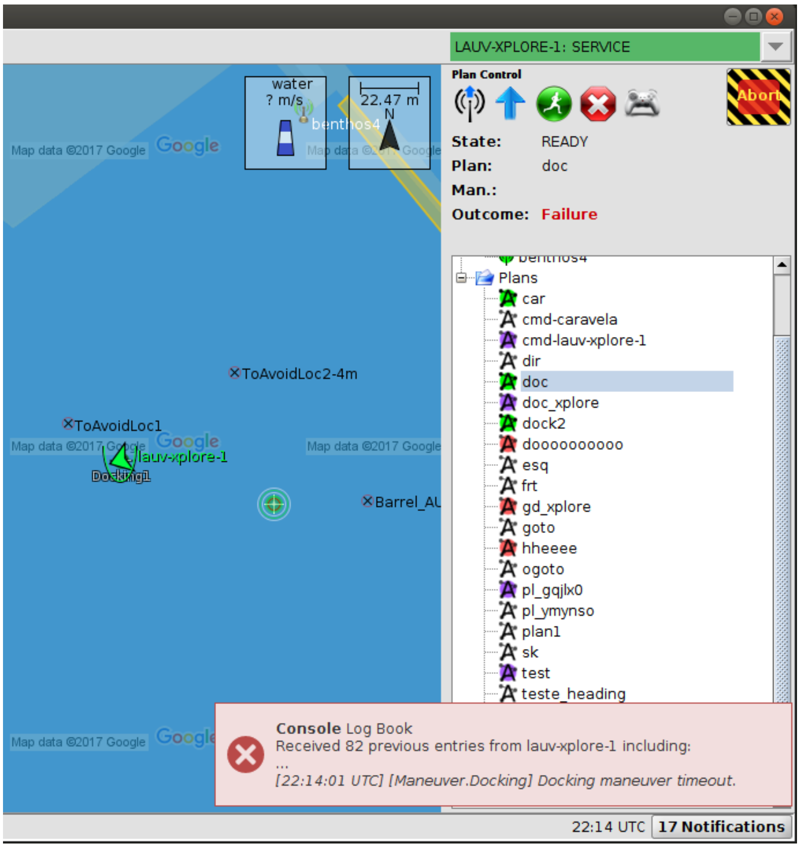


Figure 6.20: Timeout Abort

## 6.6 Conclusions

In this chapter we presented the results based on the approach defined in the previous chapter.

It was shown the test plan to test the efficiency of the docking maneuver developed. 6.1.

The result of problem 1 was shown in 6.2 showing the docking maneuver implemented in Neptus software.

The problems 3 and 4 6.3 which represent the Station Keeping and Goto Waypoint where shown.

To avoid collisions and proceed with the docking, the implementation for the problems 2, 5 and 6 where described in 6.4 where the four particular cases where presented in 6.4.2. A resume of the Homing and Docking sequence was shown in 6.4.2.5.

Finally, the abort situations are explained in 6.5.

# Chapter 7

# Conclusions and Future Work

## 7.1 Autonomous docking systems

This report presents a strategy to perform docking between an Autonomous Underwater Vehicle (AUV) and an Autonomous Surface Vehicle (ASV) *Caravela*.

The solution founded is a formulation of multiple controllers based on the LSTS software toolchain. The first controller, based on *Neptus*, serves as an interface between the vehicles and the operator that wants to perform the docking maneuver. Here, when the operator makes the maneuver plan, he will place all the parameters to perform the docking, such as the position to perform the maneuver, the function of each vehicle, the target vehicle and the number of retries when the docking maneuver fails.

It was also implemented threes *IMC* messages, the serve as a bridge between *Neptus-Dune* and *DUNE (ASV) - DUNE (AUV)*. This messages are the *Docking* message, which is the one that sent the information from *Neptus* to *DUNE*.

the *DockingState* message, which is the one sent between *DUNEs* running on the vehicles and indicates the maneuver state, availability to perform the docking, the function of the vehicle that is sending the message, and it's system name.

The *DockingUSBL* message serves to send the position of the AUV to *Caravela* so that it could calculate range and bearing between the two vehicles, like what happens with a real USBL modem.

It was also implemented in *DUNE* several sub-maneuvers such as Goto and Station Keeping maneuver, multiple controllers, with one managing the docking sequence phases has a *supervisor*, another two that control the docking and homing sequence.

In the homing sequence, it was defined that it would have a 3 point or 2 point approach, given the relative position between the ASV and the AUV. With this, collisions between vehicles are avoided. It was also implemented in *DUNE* the *Abort* sequence, which have multiple situations to be applied, as for example, when the ASV has to stop the homing or docking sequence in order to let the AUV correct its position.

Every tests where made in the simulation environment existing in *Neptus* for vehicles. Although the this was not tested in real life systems the implementation was made so that it could work after some minor changes on that systems.

## 7.2 Future Work

Given the work implemented there are a series of considerations to take as future work:

First, all the implementation made with the LSTS toolchain should be tested in real-life vehicles. Prior to that, some optimization could be done to the code, taking into account the results obtained from such experimentation.

A tracking logarithm associated with to cameras to have stereoscopic vision could be implemented in order to have a smoother docking maneuver. The ASV would track the AUV, calculate the distance between them and correct its actuation based on that information.

Although this dissertation intents to perform a full docking maneuver sequence, it is not possible with the mechanical interface, which could latch and lock the vehicles in place, performing data and power transfer, as stated before.

Given that there is not definition of what happens after the docking is completed successfully, besides data and power transfer, the ASV could also serve as a search and rescue vehicle that that would return AUV that would be at the surface of the water and could not be autonomously retrieved.

Also, a different approach could be implemented that would have only a few modifications which would be the retrieve and drop of objects that are not self propulsion capable. This could be done, for example, to drop scientific buoys.

# References

[1] F Lobo Pereira and J Borges De Sousa. Coordinated control of networked vehicles: An autonomous underwater system. *Automation and remote control*, 65(7):1037–1045, 2004.

[2] Hanumant Singh, James G Bellingham, Franz Hover, S Lemer, Bradley A Moran, Keith Von der Heydt, and Dana Yoerger. Docking for an autonomous ocean sampling network. *IEEE Journal of Oceanic Engineering*, 26(4):498–514, 2001.

[3] Roger Stokey, Mike Purcell, Ned Forrester, Tom Austin, Rob Goldsborough, Ben Allen, and Chris von Alt. A docking system for remus, an autonomous underwater vehicle. In *OCEANS'97. MTS/IEEE Conference Proceedings*, volume 2, pages 1132–1136. IEEE, 1997.

[4] Robert S McEwen, Brett W Hobson, Lance McBride, and James G Bellingham. Docking control system for a 54-cm-diameter (21-in) auv. *IEEE Journal of Oceanic Engineering*, 33(4):550–562, 2008.

[5] Michael D Feezor, F Yates Sorrell, Paul R Blankinship, and James G Bellingham. Autonomous underwater vehicle homing/docking via electromagnetic guidance. *IEEE Journal of Oceanic Engineering*, 26(4):515–521, 2001.

[6] Jonathan Evans, Paul Redmond, Costas Plakas, Kelvin Hamilton, and David Lane. Autonomous docking for intervention-auvs using sonar and video-based real-time 3d pose estimation. In *Oceans 2003. Proceedings*, volume 4, pages 2201–2210. IEEE, 2003.

[7] José Miguel Almeida Braga. Control of underwater vehicles on autonomous docking maneuvers. *Porto, Portugal: Faculdade de Engenharia da Universidade do Porto*, 2010.

[8] Thor I. Fossen. *Guidance and control of ocean vehicles.* John Wiley and Sons. ; Chichester : 1995, 1995. URL: http://search.ebscohost.com/login.aspx?direct=true&AuthType=ip,sso,url,uid&db=cat05428a&AN=bup.000421204&lang=pt-br&site=eds-live&scope=site&authtype=sso.

[9] A. J. Healey. ME4823 class notes: Dynamics and control of mobile robotic vehicles. *Naval Postgraduate School, Monterey, California*, 2009.

[10] E.V. Lewis, Society of Naval Architects, and Marine Engineers (U.S.). *Principles of naval architecture*. Principles of Naval Architecture. Society of Naval Architects and Marine Engineers, 1989. URL: https://books.google.pt/books?id=bWwZAQAAIAAJ.

[11] Guowei Cai, Ben M. Chen, and Tong Heng Lee. *Coordinate Systems and Transformations*, pages 23–34. Springer London, London, 2011. URL: http://dx.doi.org/10.1007/978-0-85729-635-1_2, doi:10.1007/978-0-85729-635-1_2.

[12] National Imagery and Mapping Agency. Department of defense world geodetic system 1984: its definition and relationships with local geodetic systems. Technical Report TR8350.2, National Imagery and Mapping Agency, St. Louis, MO, USA, January 2000. URL: http://earth-info.nga.mil/GandG/publications/tr8350.2/tr8350_2.html.

[13] Thomas A. Henzinger, Peter W. Kopke, Anuj Puri, and Pravin Varaiya. What's decidable about hybrid automata? In *Proceedings of the Twenty-seventh Annual ACM Symposium on Theory of Computing*, STOC '95, pages 373–382, New York, NY, USA, 1995. ACM. URL: http://doi.acm.org/10.1145/225058.225162, doi:10.1145/225058.225162.

[14] Anuj Puri and Pravin Varaiya. Decidable hybrid systems. In *In Hybrid Systems III, LNCS 1066*, pages 413–423. Springer, 1996.

[15] A. Deshpande and P. Varaiya. Information structures for control and verification of hybrid systems. In *American Control Conference, 1995. Proceedings of the*, volume 4, pages 2642 –2647 vol.4, jun 1995.

[16] João Hespanha. ECE229 class notes: Hybrid and switched systems, 2005. Available in http://www.ece.ucsb.edu/~hespanha/ece229/, Last time visited: April 22nd 2010.

[17] A. Deshpande and P. Varaiya. Viable control of hybrid systems. In *Decision and Control, 1996., Proceedings of the 35th IEEE*, volume 2, pages 1196 –1201 vol.2, dec 1996. doi:10.1109/CDC.1996.572649.

[18] Rajeev Alur, Costas Courcoubetis, Thomas A. Henzinger, and Pei Hsin Ho. *Hybrid automata: An algorithmic approach to the specification and verification of hybrid systems*, pages 209–229. Springer Berlin Heidelberg, Berlin, Heidelberg, 1993. URL: http://dx.doi.org/10.1007/3-540-57318-6_30, doi:10.1007/3-540-57318-6_30.

[19] Anuj Puri and Pravin Varaiya. Decidability of hybrid systems with rectangular differential inclusion. In *CAV '94: Proceedings of the 6th International Conference on Computer Aided Verification*, pages 95–104, London, UK, 1994. Springer-Verlag.

[20] R. Stokey, B. Allen, T. Austin, R. Goldsborough, N. Forrester, M. Purcell, and C. von Alt. Enabling technologies for remus docking: an integral component of an autonomous ocean-sampling network. *Oceanic Engineering, IEEE Journal of*, 26(4):487 –497, oct 2001. doi:10.1109/48.972082.

[21] B. Allen, T. Austin, N. Forrester, R. Goldsborough, A. Kukulya, G. Packard, M. Purcell, and R. Stokey. Autonomous docking demonstrations with enhanced remus technology. *OCEANS 2006*, pages 1 –6, sept. 2006. doi:10.1109/OCEANS.2006.306952.

[22] Steve Cowen, Susan Briest, and James Dombrowski. Underwater docking of autonomous undersea vehicles using optical terminal guidance. In *OCEANS'97. MTS/IEEE Conference Proceedings*, volume 2, pages 1143–1147. IEEE, 1997.

[23] Pan-Mook Lee, Bong-Hwan Jeon, and Chong-Moo Lee. A docking and control system for an autonomous underwater vehicle. In *OCEANS'02 MTS/IEEE*, volume 3, pages 1609–1614. IEEE, 2002.

[24] Anthony J Healey and David Lienard. Multivariable sliding mode control for autonomous diving and steering of unmanned underwater vehicles. *IEEE journal of Oceanic Engineering*, 18(3):327–339, 1993.

[25] Jin-Yeong Park, Bong-Huan Jun, Kihun Kim, Pan-Mook Lee, Jun-Ho Oh, and Yong-Kon Lim. Improvement of vision guided underwater docking for small auv isimi. In *OCEANS 2009, MTS/IEEE Biloxi-Marine Technology for Our Future: Global and Local Challenges*, pages 1–5. IEEE, 2009.

[26] Keith Vickery. Acoustic positioning systems. a practical overview of current systems. In *Autonomous Underwater Vehicles, 1998. AUV'98. Proceedings of the 1998 Workshop on*, pages 5–17. IEEE, 1998.

[27] Wen-Hui Cheng. A study of increasing the precision of navigation position for submerged body. *Ocean engineering*, 31(5):693–707, 2004.

[28] Li Zhou. *A precise underwater acoustic positioning method based on phase measurement*. PhD thesis, University of Victoria, 2010.

[29] G. Remmers, R. Taylor, P. Palo, and R. Brackett. Mobile offshore base: a seabasing option. *Proceedings of Third International Workshop on Very Large Floating Structures*, 1-7, 1999.

[30] Anouck Girard, J. Karl Hedrick, and J. Borges de Sousa. A hierarchical control architecture for mobile offshore bases. *Marine Structures*, 13(4-5):459 – 476, 2000. URL: http://www.sciencedirect.com/science/article/B6V41-41SK9J9-J/2/d0ce205dbea6db57b5c64fdb063698d4, doi:DOI:10.1016/S0951-8339(00)00035-6.

[31] A. Girard, J. Borges de Sousa, and K. Hedrick. Coordinated control of agent formations in uncertain, dynamic environments. In *Proceedings of the European Control Conference*, Porto, Portugal, 9 2001.

[32] J. Borges de Sousa, J. Karl Hedrick, Anouck Girard, and William Webster. A ship maneuvering control framework. In *Proceedings of the International Conference on Offshore Mechanics and Arctic Engineering - OMAE*, volume 5, pages 111 – 119, Rio de Janeiro, Brazil, 2001. Control architecture;Hybrid systems;.

[33] Jorge Estrela da Silva, Bruno Terra, Ricardo Martins, and Joao Borges de Sousa. Modeling and simulation of the lauv autonomous underwater vehicle. 2007.

[34] Anthony J Healey, DB Marco, et al. Slow speed flight control of autonomous underwater vehicles: Experimental results with nps auv ii. In *The Second International Offshore and Polar Engineering Conference*. International Society of Offshore and Polar Engineers, 1992.

[35] M. Blanke. *Ship Propulsion Losses Related to Automatic Steering and Prime Mover Control*. Technical University of Denmark, 1981. URL: https://books.google.pt/books?id=g_XgNAAACAAJ.

[36] KSM Davidson and L Schiff. Turning and course keeping qualities of ships. 1946.

[37] Kensaku Nomoto and Kenshi Taguchi. On steering qualities of ships (2). *Journal of Zosen Kiokai*, 1957(101):57–66, 1957.

[38] Jason Carbaugh, Datta N Godbole, and Raja Sengupta. Safety and capacity analysis of automated and manual highway systems. *Transportation Research Part C: Emerging Technologies*, 6(1):69–99, 1998.

[39] Claire Tomlin, George Pappas, Jana Košecká, John Lygeros, and Shankar Sastry. Advanced air traffic automation: A case study in distributed decentralized control. *Control Problems in Robotics and Automation*, pages 261–295, 1998.

[40] Riccardo Schiavi, Antonio Bicchi, and Fabrizio Flacco. Integration of active and passive compliance control for safe human-robot coexistence. In *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, pages 259–264. IEEE, 2009.