DESIGN AND CONTROL OF AN UNMANNED SURFACE VEHICLE FOR OIL SPILL CLEANUP

A thesis submitted in partial fulfillment of the requirements for the award of the degree

B.Tech

in

Mechanical Engineering

By

Gudapati Nitish (111117036)

Devarasetty Sasi Preetham (111117030)

Athithya Kumar NB (111117019)

DEPARTMENT OF MECHANICAL ENGINEERING
NATIONAL INSTITUTE OF TECHNOLOGY
TIRUCHIRAPPALLI – 620015

MAY 2021

BONAFIDE CERTIFICATE

This is to certify that the project titled **DESIGN AND CONTROL OF AN UNMANNED SURFACE VEHICLE FOR OIL SPILL CLEANUP** is a bonafide record of the work done by:

GUDAPATI NITISH (111117036)

DEVARASETTY SASI PREETHAM (111117030)

ATHITHYA KUMAR NB (111117019)

in partial fulfillment of the requirements for the award of the degree **BACHELOR OF TECHNOLOGY** in **MECHANICAL ENGINEERING** at the **NATIONAL INSTITUTE OF TECHNOLOGY, TIRUCHIRAPPALLI**, during the year 2020-2021.

26/5/21

Dr. T. Ramesh
(Project Guide)

Dr. AR. Veerappan
(Head of the Department)

Project Viva-voce held on: _____

Internal Examiner

External Examiner

ABSTRACT

Oil spills seem to be one of the main reasons for sea and river pollution caused by human beings. Most of the time, oil is accidentally spilled during transportation, storage, or due to equipment malfunctioning. The marine ecosystem is at great risk as oil spills have detrimental effects on aquatic life, disturbing the biological balance. These effects have attracted several researchers from different disciplines involving biology, petroleum, environment, marine, chemical, and materials engineering. Although there are various techniques being currently used, developing an effective, sustainable, and ergonomic oil cleanup approach is still a challenging task due to the high cost and environmental impact involved in the current practices.

In this paper, we discuss the design of a unique Unmanned Surface Vehicle (USV) which facilitates the versatile and agile motion of the robot. The mathematical modeling, including the kinematics and dynamics of the robot, and Proportional-Integral-Differential control is presented. The outcome of this work is a proposal of a technological solution involving the design and control of the robot with a skimmer arrangement for efficient oil recovery. Further, enhancements can be made in the form of path planning, swarm intelligence, and centralized control.

Keywords : Oil-spill, Water body pollution, Robots, Unmanned Surface Vehicle (USV).

ACKNOWLEDGEMENTS

We would like to express our deepest gratitude to the following people for guiding us through this course and without whom this project and the results achieved from it would not have reached completion.

Dr. T. Ramesh, Associate Professor, Mechanical Engineering, for helping and guiding us in the course of this project. Without his guidance, we would not have been able to successfully complete this project. His directions and suggestions helped us immensely to see the areas to focus and also face the difficulties due to the pandemic with confidence. His patience and genial attitude is and always will be a source of inspiration to us.

Dr. AR Veerappan, the Head of the Department, Mechanical Engineering, for allowing us to avail the department resources and for his motivation.

We are also thankful to the faculty and staff members of the Mechanical Engineering, our individual parents and our friends for their constant support and help.

TABLE OF CONTENTS

Title	Page No.
ABSTRACT	i
ACKNOWLEDGEMENTS	ii
TABLE OF CONTENTS	iii
LIST OF TABLES	v
LIST OF FIGURES	vi
CHAPTER 1 INTRODUCTION	1
1.1 Classification of Oil Spills	2
1.2 Properties of Oil Spill and Oil Slick	3
1.2.1 Physical Characteristics	3
1.2.2 Chemical Characteristics	3
1.3 Methods of Oil Spill Cleanup	4
1.3.1 Mechanical Methods	4
1.3.2 Chemical Methods	6
1.3.3 Thermal Methods	7
1.3.4 Bioremediation	8
CHAPTER 2 LITERATURE REVIEW	9
2.1 Swarm of Robots used for oil spill cleanup	9
2.2 Seaswarm	10
2.3 Protei	10
2.4 Bio-cleaner based on *Roomba*	11
CHAPTER 3 DESIGN OF THE USV	12
3.1 Oil extraction and Thruster configuration	12

3.2 Dimensions and Materials 13
CHAPTER 4 MATHEMATICAL MODELLING 15
4.1 Conventions and Notations 15
4.2 Kinematics 15
4.3 Dynamics 17
CHAPTER 5 CONTROL OF THE USV 24
5.1 Plant Model 24
5.2 PID Controller 25
5.2.1 Proportional Controller 26
5.2.2 Proportional-Integral (PI) Controller 26
5.2.3 Proportional-Integral-Derivative (PID) Controller 26
5.3 Implementation 26
CHAPTER 6 RESULTS AND CONCLUSION 28
6.1 Simulation Model 28
6.2 Response of the USV to Different Trajectories 29
6.2.1 Circle 29
6.2.2 Square 30
6.2.3 Sine Curve 31
6.3 Conclusion 32
6.4 Future Work 32
APPENDIX A Code Attachments 33
A.1 Main Program 33
A.2 Control Algorithm 35
A.3 Simulation Model 39
References 40

LIST OF TABLES

3.1 Classification and oil recovery potential of skimmers [12] 12
3.2 Materials [14] 14
5.1 System response to increase in PID Gains [15] 26
6.1 Parameters for Circular Trajectory (in SI units) 30
6.2 Parameters for Square Trajectory (in SI units) 31
6.3 Parameters for Sine Curve Trajectory (in SI units) 32

LIST OF FIGURES

1.1 Processes undergone by oil after spilled onto the water bodies [5] 2
1.2 Types of Booms 5
1.3 Oleophilic Skimmers 5
1.4 Dispersant Spraying 6
1.5 Oil is solidified after being treated by solidifiers 7
1.6 Oil burning and pollutant emission. 7

2.1 The proposed robot with folded skimmer. *Source:*[9] 9
2.2 Connections between the components. *Source:*[9] 9
2.3 Swarm of Seaswarm Robots 10
2.4 Proposed Design *Source:*[10] 10
2.5 Protei Prototype *Source:*[11] 11
2.6 *Romba* Floating unit *Source:*[11] 11

3.1 Isometric view of the USV 13
3.2 Thruster Configuration 13
3.3 Front and side views of the USV 13

4.1 Convention 15
4.2 Vector Addition 18

5.1 Plant or Process 25
5.2 Plots of transfer functions with time 25
5.3 System with PID Controller 27
5.4 Step Response of the Controllers (Amplitude vs Time plot) 27

5.5 Ramp Response of the Controllers (Amplitude vs Time plot) 27

6.1 Simulation Model 28

6.2 Circular Trajectory Response 29

6.3 Square Trajectory Response 30

6.4 Sine Curve Trajectory Response 31

CHAPTER 1

INTRODUCTION

Oil spill disasters not only affect marine life, but also pollute the atmosphere and have a detrimental effect on human beings. Environmental pollution, the impact caused by the oil spill and its appearance on the ocean surface are critical to highlight among the various toxic substances that are discharged into the oceans. [1]

During 1970 to 2010, about 5.71 million heaps of oil were spilled due to oil tanker accidents. These incidents severely affected the marine ecosystem, water quality, tourism and recreational activities. [2, 3].

Oil-slick formation depends on many factors, some of which are, [4]

- Weather
- Rate of spreading on different water bodies
- Rafting of layer in sea water
- Vaporization into air
- Bio-degradation
- Formation of emulsions
- Change in viscosity & density
- Interfacial tension forces

Oil clearing strategies including bioremediation, mechanical methods, in-situ methods (burning), utilization of solidifiers and dispersants are examined to be valuable.

Appropriate methods can be selected based on the type and quantity of oil spilled, and environmental conditions. The variation, transformation, and alteration process of oil and its interaction with the water in seas and oceans under natural conditions are depicted in figure 1.1.

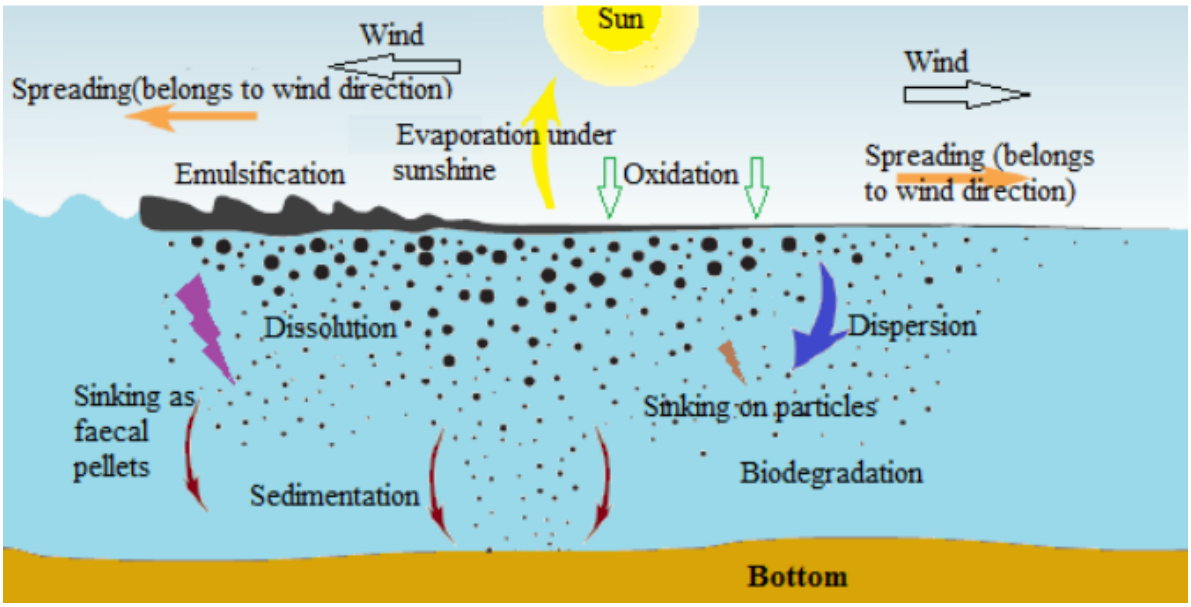


Figure 1.1: Processes undergone by oil after spilled onto the water bodies [5]

1.1 Classification of Oil Spills

Oil spills in marine environment are classified into three categories based on the size and way of formation:

1. **The First Class** - Small oil spills and slicks that are caused by mooring of ships or vessels operating at the port, or by discharge or leakage of oil from heavy machinery.

2. **The Second Class** - Oil spills and slicks due to collisions, explosions, and other damages. These oil spills majorly occur in bigger zones and have unfavourable effects on the ocean ecosystem.

3. **The Third Class** - As an attempt to save money, waste oil is dumped purposefully into the ocean. The pouring of oil into the ocean is illegal and the concerned may be subjected to penalty or imprisonment.

Statistics show that every year two to four Second and Third class oil spills take place worldwide. The majority of the oil spills that took place in history have been calamitous, which tipped the ecological balance and caused severe socio-economic losses.

1.2 Properties of Oil Spill and Oil Slick

1.2.1 Physical Characteristics

The physical properties of oil like thickness, density, consistency, solubility in water, surface energy and so forth influence the oil film and its boundaries.

Density of most oils is usually less than seawater, causing them to float and lie flat on the surface. This also results in the spreading and expansion of oil horizontally. The oil, having lower density, experiences increased evaporation of the lighter compounds, leaving the heavier compounds to sink into the ocean. These heavier compounds interact with water and other substances in the ocean to form noxious sedimentation on the ocean bed.

Rate of oil spreading can be found by using **oil viscosity** as a parameter. Furthermore, an increase in temperature results in a reduction of the viscosity and density of the liquids, thereby increasing the oil spread rate.

Solubility of oil in water can be related to the formation of pollutants. Bio-remediation is low and it depends on the temperature of the water and chemical structure of hydrocarbons involved. Oil's solubility in water is usually around 30 mg/l [6].

Surface tension of oil varies inversely with the temperature. Consequently, oil spreads faster on water at higher temperatures. This directly affects the spreading ability in the absence of winds or water currents as well.

1.2.2 Chemical Characteristics

Hydrocarbons, which form major composition of oils, dominate their chemical characteristics. In addition, oil consists of non-hydrocarbons like non-metals like sulphur, oxygen and nitrogen along with traces of metals.

Oils can be separated into saturated (alkanes), unsaturated and aromatic hydrocarbons mainly along with a minimal quantity of other refined products [7]. Alkanes are difficult to dissolve in seawater. Aromatic hydrocarbons are serious pollutants as they are considered as potential carcinogens. Nearly half of the composition of crude oil is cyclic hydrocarbons and then 30% are alkanes, 15% are aromatics and the remaining are other non-HC compounds. Refined oil consists of unsaturated hydrocarbons which are the products of catalytic cracking.

1.3 Methods of Oil Spill Cleanup

Oil spill and slick cleanup remains to be challenging as it is impossible to clear the discharged oil into the seawater completely. Modern recovery and recycle strategies include mechanical methods, using chemicals, thermal methods like burning and organic techniques like bioremediation.

1.3.1 Mechanical Methods

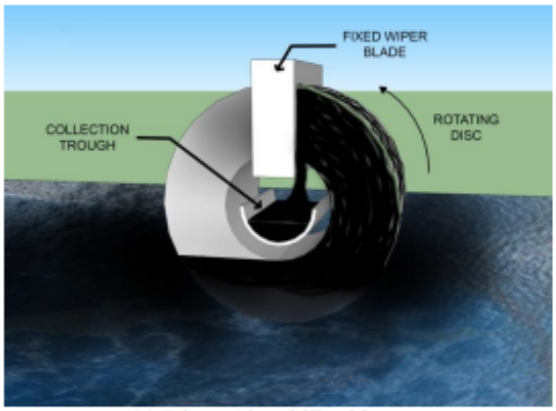
Mechanical or physical methods help by limiting and controlling the spread of spilled oil without compromising the oil's chemical and physical properties. Barriers such as booms and skimmers, or other adsorbent materials can be used for controlling the same [8].

1. **Booms**: A Boom is a setup used to control and prevent the spread of spilled oil. In any case, the operation of booms relies upon the boom configuration as well as unequivocally influenced by the characteristics of wind, its course, speed and height of waves. After restricting the oil movement using these barriers, the oil can be recovered through skimmers or other techniques.

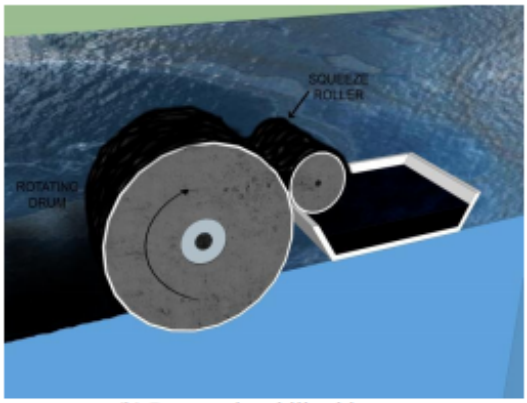
    - **Fence booms** are flexible floating systems that are used to control the vertical movement of oil that happens with the waves. Normally, only 40% of fence boom lies above the water surface. Some of the benefits of fence booms include minimal weight and storage space, resistance to corrosion. Hence, they are efficient on calm waters.
    - **Curtain booms** have large circular chamber filled with foam to float on the water surface without being pervious. Although curtain booms are highly flexible and more reliable when compared to fence booms, they consume more storage space and are difficult in cleanups.
    - **Resistant booms** are manufactured using fireproof materials. This gear can collect and maintain the spilt oil to the most appropriate temperature and burn them.

Figure 1.2: Types of Booms

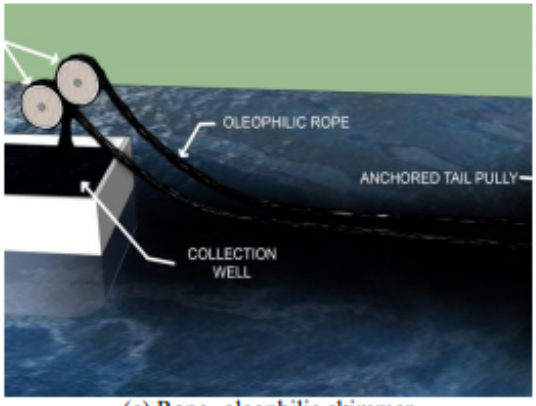
2. **Skimmers**: These are basically used as recycle tool to collect the oil spill and slick that was circumscribed and limited by the booms. The collected oil will have no change in its properties. Skimmers are categorized as weir skimmers, elevating skimmers, submersion skimmers, vacuum skimmers, oleophilic skimmers among many.



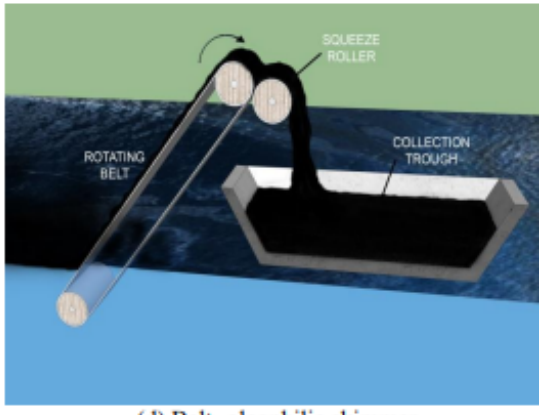
(a) Disc - Oleophilic skimmer



(b) Drum- oleophilic skimmer



(c) Rope- oleophilic skimmer



(d) Belt- oleophilic skimmer

Figure 1.3: Oleophilic Skimmers

3. **Adsorbent materials**: The main purpose of adsorbent materials is to recover the spilled oil at the final stage of the cleanup process. It is usually done after the majority of oil is recuperated by other physical methods like skimmers of high oil

adsorbing capacity. The adsorbent materials are used to convert the liquid into a semisolid in order to remove the spilled oil. There are three types of adsorbent materials including natural organic materials, inorganic absorbent materials and synthetic materials.

Some of the advantages of physical techniques include the ability to recover majority of oils, non-destructive, simple and efficient as a final cleanup method. Disadvantages include expense, complexity, need to be used before emulsification and inability to use without being assisted by technological devices. Their performance is also affected by the weather conditions and the collected oil must be treated before usage. The use of synthetic sorbents also raises concerns about their biodegradability.

1.3.2 Chemical Methods

Chemical methods have the capability to change physical and chemical properties of the spilt oil. Hence, they are used in combination with physical methods. These methods use dispersants and solidifiers as the main chemicals to control the spilt oil.

1. **Dispersants**: A dispersant is a mixture of solvent and emulsifiers which aid the dispersion of oil into small droplets. Dispersants work by reducing the interfacial surface tension between the water and spilled oil. This promotes dispersion and bio-degradation of oil in the ocean.

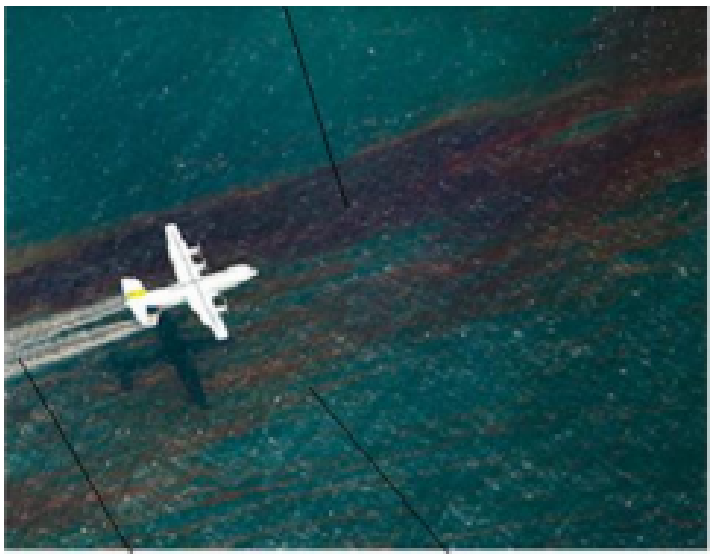


Figure 1.4: Dispersant Spraying

Some of their advantages include high cleanup capacity to nearly 90% and potential to work effectively on rough seas. Their disadvantages include composition of toxic compounds, ineffectiveness in calm seawater and high cost.

2. **Solidifiers**: Solidifiers constitute of dry powdered substances that can react and change the state of oil from liquid to solid for easy extraction. The principle benefit of solidifiers is that they can be used on rough seas.
   Examples of Solidifiers include Spill Green LS, Petro Lock, SmartBond HO.

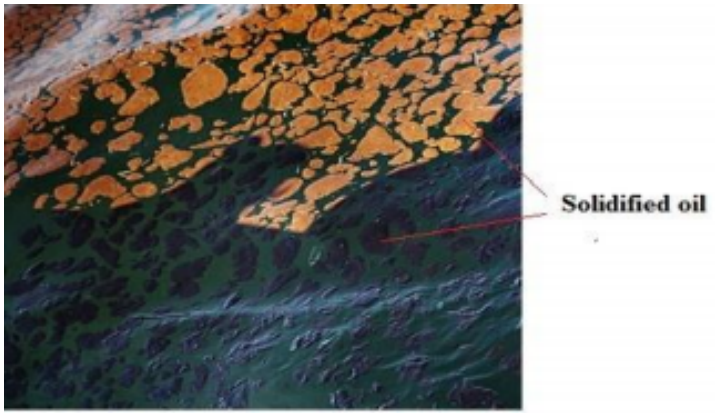


Figure 1.5: Oil is solidified after being treated by solidifiers

In comparison with physical methods, chemical methods have high efficiency, are cheaper and do not need much manpower. However, their disadvantages are that oil cannot be recovered, not effective with high viscosity oils and pose environmental concerns.

1.3.3 Thermal Methods

As shown in the figure, thermal methods like burning the oil on the surface of water is simple and hence commonly employed. This method is used to mitigate the harmful consequences of oil spills on underwater ecosystems and the marine environment.

Figure 1.6: Oil burning and pollutant emission.

About 100-300 tons/hour of spilled oil can be removed by thermal methods. This method has been commonly used in US, Europe and Canada, since 1960s, to clear the spilt oil on water bodies and on ice or snow which are caused due to ship accidents.

Thermal or burning method is highly preferred on calm waters where the spill is sufficiently wide and thick to sustain the flames and combustion.

The main disadvantage of this method is the possible occurrence of secondary fires, its deteriorating effect on health and the environmental hazard caused due to the by-products of combustion. Pollutants such as CO_x, SO_x, NO_x, particulate matter, etc. are emitted as by-products of the combustion of oil.

1.3.4 Bioremediation

Biodegradation involves micro-organisms like yeast, fungi and bacteria that degrade, decompose and then convert the chemical compounds into their energy source thereby restoring and maintaining the environmental quality.

Numerous enzymatic microorganisms show the capacity to disintegrate hydrocarbons in gasoline, crude oil, or diesel.

In addition to aromatic content (such as benzene, toluene or low molecular weight xylene), alkanes with 10 to 26 carbon atoms also degrade faster. As the structure becomes complex, the microorganisms face difficulty in degrading the oil. The biodegradation of the spilt oil is impacted by several parameters like concentration of the oil, nutrient availability and temperature.

Nutrients (such as phosphorus, nitrogen and dissolved oxygen) are essential for the metabolism of micro-organisms. If the concentration of spilt oil is high, its biodegradation takes around 2 to 4 weeks. This method, being cheaper and efficient, is suitable for all weather conditions and the end products of this method are only CO_2, H_2O.

CHAPTER 2

LITERATURE REVIEW

2.1 Swarm of Robots used for oil spill cleanup

In this study, the design and application of a surface vehicle having a semi-oval shaped skimmer of diameter four meters was proposed. This skimmer helps to circumscribe the oil spill, lift it off the water and then help to move the spilt oil to some other location for easier clean up and better skimming [9].

A collective system of proposed robots act as swarm robots which meets the specifications for any swarm robotic group in navigation, sensing, mobility and localization. Each robot in the swarm was equipped with a Digital compass, motor driver, micro-controller, voltage regulator, LCD, transceiver module, GPS module, etc.

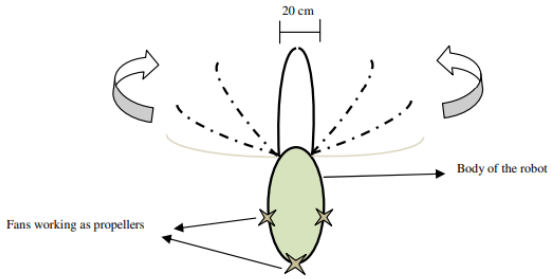


Figure 2.1: The proposed robot with folded skimmer. *Source:*[9]

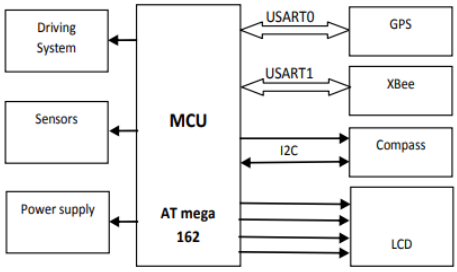


Figure 2.2: Connections between the components. *Source:*[9]

The robot navigates with the help of GPS and an on-board digital compass for maintaining its heading. Based on the GPS data and the digital compass, the orientation and the oil spill's center distance were calculated by a *dispatching algorithm*. Then via wireless transmission signals that contain GPS coordinates of center of oil spill, the boundary of the working area is covered by signaling a swarm of robots [9].

2.2 Seaswarm

Seaswarm consists of a fleet of cost effective oil extracting robots. Seaswarm proposes a new system of autonomous swarm surface robots that removes oil and skims the ocean. It collects oil and propels using a conveyor belt powered by a photovoltaic cell. This belt is made of a thin nanowire mesh which is capable of taking up to 20 times of oil by weight [10].

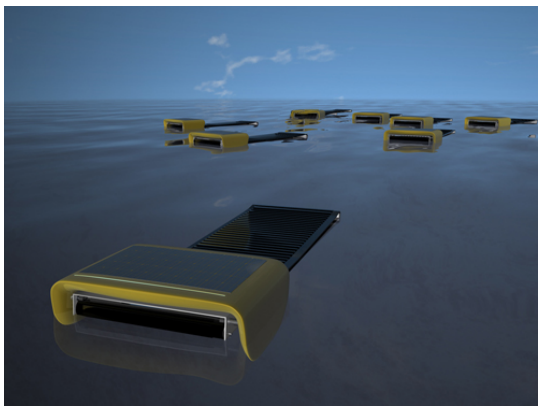


Figure 2.3: Swarm of Seaswarm Robots

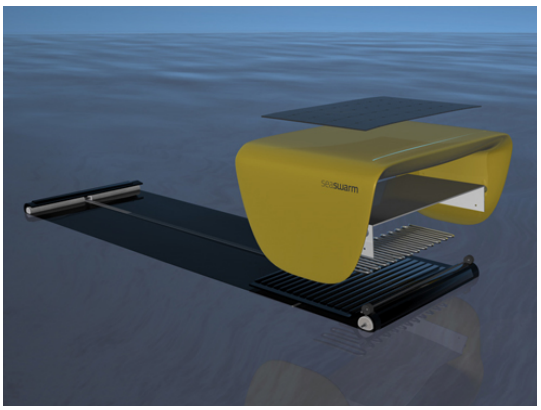


Figure 2.4: Proposed Design *Source:*[10]

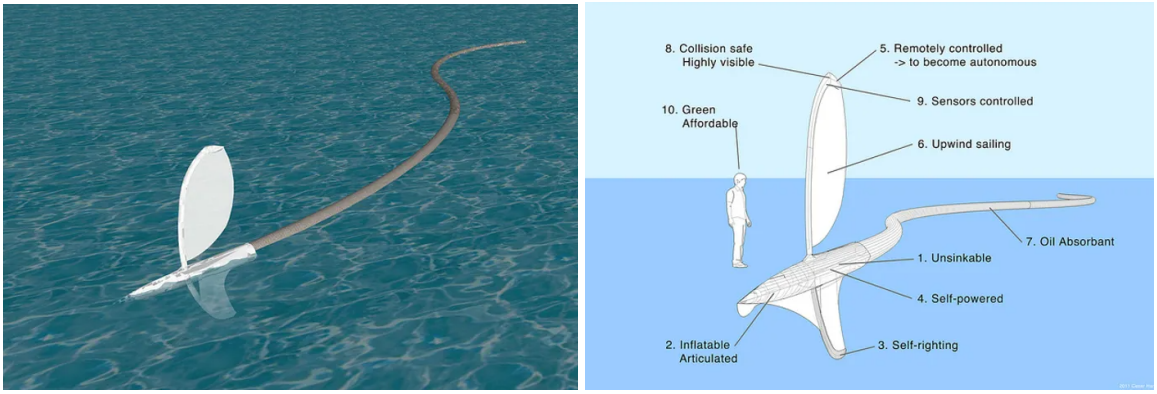
The flexible photovoltaic conveyor belt absorbs oil by gently rolling over the surface of the ocean and deflects water owing to its hydrophobic nature. Being a swarm technology, it works collectively. In order to create an autonomous organized system of 'swarm', this fleet of robots communicate their location and motion via GPS and WiFi signals.

Seaswarm detects the spill's contour and moves inward until it completely covers a single patch of ocean. The hydrophobic material skims lipids like oils but deflects water and then it is heated to remove the oil. To avoid the repeated trips back to shore for dropping off the collected oil, the oil is "digested" locally in each Seaswarm itself [10].

2.3 Protei

Protei is an open source fleet of oil spill cleaning robot drones. It also acts as pollution collecting, sailing drone.

Protei utilizes wind power in order to propel itself. It is designed with a long oil-absorbent 'tail' that pulls oil in upwind. A number of such robots can be wirelessly controlled from the coast or work autonomously as a swarm [11].



(a) (b)

Figure 2.5: Protei Prototype *Source:*[11]

2.4 Bio-cleaner based on *Roomba*

Roomba is cleaning equipment which runs around, cleaning the floors of homes. The idea proposed is of a *Bio-Cleaner system*, based on the technology of Roomba, which would be an efficient method in oil-Spill cleanup and recovery. The Roomba and its working principle are shown in the Figure below [11].

Figure 2.6: *Romba* Floating unit *Source:*[11]

The bot is deployed from the helicopter or boat. It then tracks the oil spills and centralizes on removing the spill in that zone. The robot is equipped with a compartment full of bacteria that digests the oil. In this digestive chamber, the bacteria degrade the oil and return clean ocean water [11].

CHAPTER 3

DESIGN OF THE USV

3.1 Oil extraction and Thruster configuration

The motivation of this study is to design an Unmanned Surface Vehicle that utilizes one of the physical methods of oil-spill removal - skimmers. We propose to use an Oleophilic skimmer with 0.2 to 50 m^3/h range of oil recovery rate which is found to be higher than other conventional physical methods with nearly 75-95% oil recovered as shown in Table 3.1.

Types of Skimmers	**Principle of Operation**	**Oil Recovery Rate (m^3/h)**	**Oil Recovery Efficiency (%)**
Oleophilic	Oil on the water surface adheres to an oleophilic surface thereby cleaning the surface.	0.2 to 50	75 to 95
Weir	Draining the oil using gravity based skimmers	0.2 to 100	20 to 90
Elevating	Usage of conveyors for extracting oil to the recovery area	1 to 20	10 to 40
Submersion	Usage of an inclined belt to collect the oil into a tank.	0.5 to 80	70 to 95
Suction or Vacuum	Oil is removed through suction by creating a pressure difference	0.3 to 40	3 to 90
Vortex or Centrifugal	Water is separated from the mixture by vortex force, as oil is less dense than water.	0.2 to 10	2 to 20

Table 3.1: Classification and oil recovery potential of skimmers [12]

For the ease of collection and extraction of oil, we have come up with a belt drive mechanism, in which the belt surface is coated with an oleophilic material. This belt drive is to be mounted on the robot and driven by a motor. As the robot floats, the belt stays partially immersed and rotates over the rollers to absorb the oil. With the help of another roller, as shown in Fig. 3.1, the oil can be squeezed out of the belt and collected in the tank.

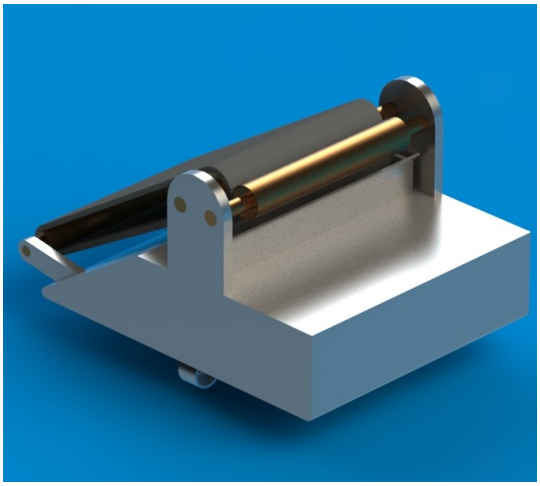


Figure 3.1: Isometric view of the USV

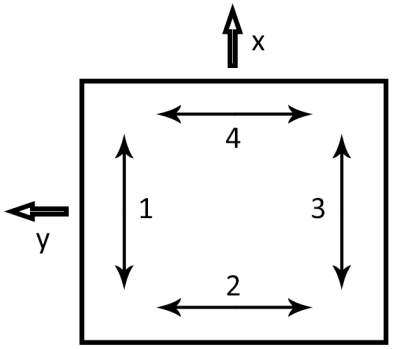


Figure 3.2: Thruster Configuration

To facilitate holonomic motion of the robot, we propose the usage of 4 two-way underwater thrusters, two of them along X direction and the other two along Y direction, as shown in Fig. 3.3. These thrusters are to be mounted below the trapeziodal tank. Appropriate propeller and motor can be chosen based on the parameters calculated using the equations given in the section 3 of [13].

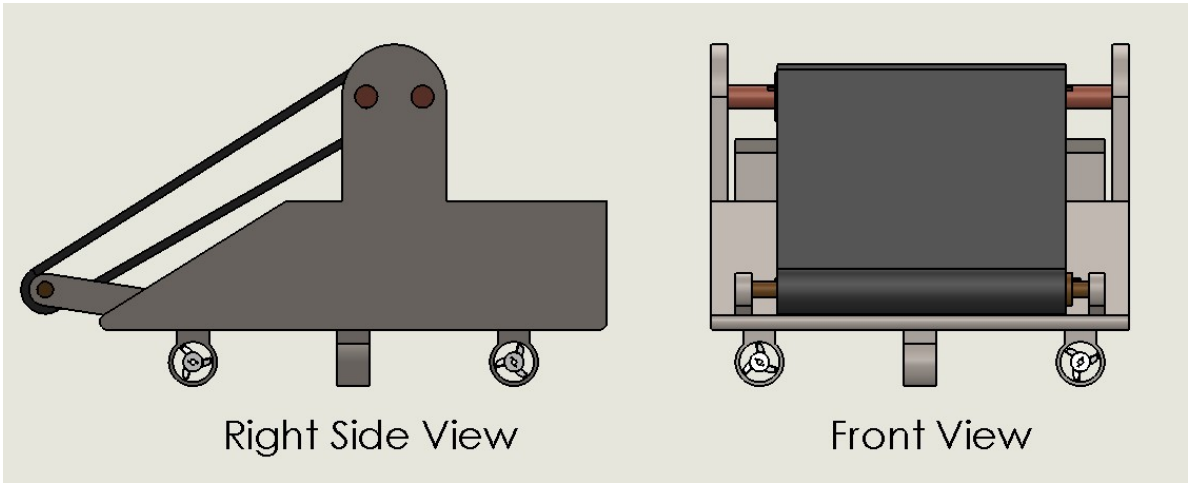


Figure 3.3: Front and side views of the USV

3.2 Dimensions and Materials

The prototype design is made using SolidWorks 2016. The main factors to be considered while designing the USV are:

- Oil collection rate
- Tank capacity

- Weight of USV

As per the above requirements, trapezoid shape is chosen for the tank to have maximum tank capacity and mount the belt drive at a comfortable orientation. The rollers and the squeezer can be installed onto the mounts that are attached to the tank. There is a collector placed beneath the roller and the squeezer for the passage of oil into the tank. The tank, mounts and the collector together form the body of the USV which can be 3D printed.

Since this is a prototype, the dimensions can be scaled to meet the practical requirements after taking into consideration of several parameters. The following are the basic dimensions of the prototype:

- Total: Length = 350mm, Width = 260mm, Height = 215mm
- Tank capacity: 4 litres
- Belt: Length = 545mm, Width = 180mm, Thickness = 5mm
- Distance between opposite thrusters = 150mm

For the USV to float on water, all the parts must have low density. At the same time, the load bearing components like rollers, squeezer and belt should have sufficient strength to withstand the load. Manufacturing these parts can be made cost effective and faster if they are 3D printed. Taking into account of all the factors, we suggest the following materials, as shown in Table 3.2

Parts	**Material**	**Density** (kg/m^3)	**Tensile Strength** (MPa)
Body	Nylon 101	1150	79.3
Rollers	Nylon 101	1150	79.3
Squeezer	Nylon 101	1150	79.3
Belt	Styrene-butadiene	940	245

Table 3.2: Materials [14]

Using these materials and considering additional components to be added such as motor, thrusters, batteries and other electronic components, the total estimated weight of the prototype is about 1.8 kg and the moment of inertia about Z axis is 0.03 kgm^2. Assuming the belt runs at a speed of 20 rpm and the oil film on the belt has a thickness of 1 mm, the oil collection rate of the prototype is approximately 0.02 m^3/h.

CHAPTER 4

MATHEMATICAL MODELLING

4.1 Conventions and Notations

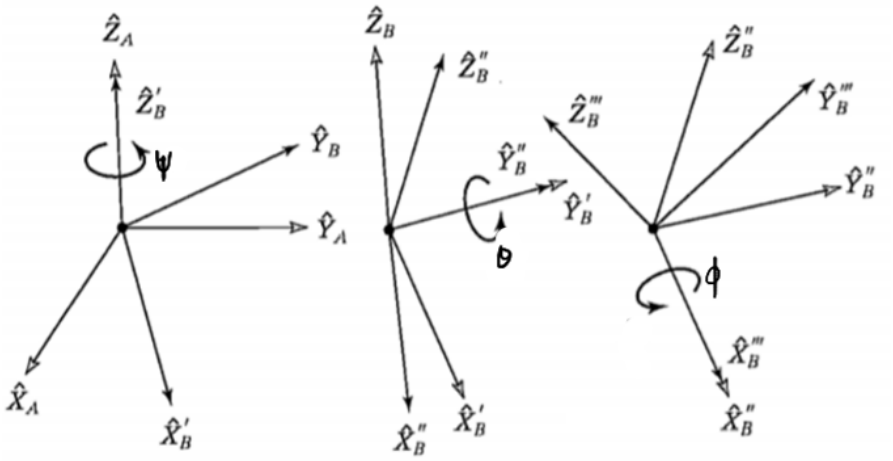


Figure 4.1: Convention

where,

A - Inertial or Earth frame of reference

B - Body Frame of reference

B' - Intermediate frame of reference after first rotation

B" - Intermediate frame of reference after second rotation

4.2 Kinematics

Transformation of Linear Velocities

Let the linear velocities in global or ground frame of reference be represented as $\dot{\zeta} = \begin{bmatrix} \dot{x} & \dot{y} & \dot{z} \end{bmatrix}^T$ and that in the local or body frame of reference be represented as $v_B = \begin{bmatrix} u & v & w \end{bmatrix}^T$.

The rotation about z-axis is given by,

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} cos(\psi) & -sin(\psi) & 0 \\ sin(\psi) & cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix}$$

Similarly, Transformation matrices for rotation about z-axis, y-axis and x-axis

respectively are given by,

$$R^A_{B'} = \begin{bmatrix} cos\psi & -sin\psi & 0 \\ sin\psi & cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}, R^{B'}_{B"} = \begin{bmatrix} cos\theta & 0 & sin\theta \\ 0 & 1 & 0 \\ -sin\theta & 0 & cos\theta \end{bmatrix}, R^{B"}_{B} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos\phi & -sin\phi \\ 0 & sin\phi & cos\phi \end{bmatrix}$$

Rotation about x, followed by y and then z axis,

$$R^A_B(\psi, \theta, \phi) = R^A_{B'}(\psi) \cdot R^{B"}_{B'}(\theta) \cdot R^B_{B"}(\phi)$$

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} c\theta c\psi & s\phi s\theta c\psi - c\phi s\psi & c\phi s\theta c\psi + s\phi s\psi \\ c\theta s\psi & s\phi s\theta s\psi + c\phi c\psi & c\phi s\theta s\psi - s\phi c\psi \\ -s\theta & s\phi c\theta & c\phi c\theta \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix} \tag{4.1}$$

Transformation of Angular Velocities

Let the angular velocities in inertial frame be represented as $\dot{\eta} = \begin{bmatrix} \dot{\phi} & \dot{\theta} & \dot{\psi} \end{bmatrix}^T$ and that in the body frame be represented as $\omega_B = \begin{bmatrix} P & Q & R \end{bmatrix}^T$
The vector $\bar{\omega}_B$ is given by,

$$\bar{\omega}_B = iP + jQ + kR = \dot{\bar{\phi}}_B + \dot{\bar{\theta}}_B + \dot{\bar{\psi}}_B \tag{4.2}$$

Representing $\dot{\bar{\psi}}$ in the body frame B,

$$\dot{\bar{\psi}}_B = R^B_{B"}(\phi) R^{B"}_{B'}(\theta) \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix}$$

Representing $\dot{\bar{\theta}}$ in the body frame B,

$$\dot{\bar{\theta}}_B = R^B_{B"}(\phi) \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix}$$

Representing $\dot{\bar{\phi}}$ in the body frame B,

$$\dot{\bar{\phi}}_B = \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix}$$

Substituting $\dot{\bar{\psi}}_B$, $\dot{\bar{\theta}}_B$ and $\dot{\bar{\phi}}_B$ in equation 4.2,

$$\omega_B = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c\phi & s\phi \\ 0 & -s\phi & c\phi \end{bmatrix} \begin{bmatrix} c\theta & 0 & -s\theta \\ 0 & 1 & 0 \\ s\theta & 0 & c\theta \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 \\ 0 & c\phi & s\phi \\ 0 & -s\phi & c\phi \end{bmatrix} \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} + \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix}$$

Simplifying the above equation, we get

$$\omega_B = \begin{bmatrix} 1 & 0 & -sin\theta \\ 0 & cos\phi & sin\phi cos\theta \\ 0 & -sin\phi & cos\phi cos\theta \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix}$$

$$\dot{\eta} = \begin{bmatrix} 1 & sin\phi tan\theta & cos\phi tan\theta \\ 0 & cos\phi & -sin\phi \\ 0 & sin\phi/cos\theta & cos\phi/cos\theta \end{bmatrix} \omega_B \tag{4.3}$$

Assuming the robot does not translate along Z axis and does not rotate about X and Y axis, given the planar configuration of the thrusters,

$$\theta = \phi = w = P = Q = 0, \qquad \dot{w} = \dot{P} = \dot{Q} = 0.$$

Substituting these values in equations 4.1 and 4.3,

$$\dot{x} = ucos\psi - vsin\psi$$

$$\dot{y} = usin\psi + vcos\psi$$

$$\dot{\psi} = R$$

4.3 Dynamics

Using Newton's laws of motion,

$$\frac{d}{dt}\int_V \rho_a(\frac{d\bar{r'}}{dt})dV = \int_V \rho_a \bar{g} dV + \int_S \bar{F} ds \tag{4.4}$$

$$\frac{d}{dt}\int_V \bar{r} \times \rho_a(\frac{d\bar{r'}}{dt})dV = \int_V \bar{r} \times \rho_a \bar{g} dV + \int_S \bar{r} \times \bar{F} ds \tag{4.5}$$

Here, $\bar{r'} = \bar{r'_p} + \bar{r}$ as shown in Fig. 4.2

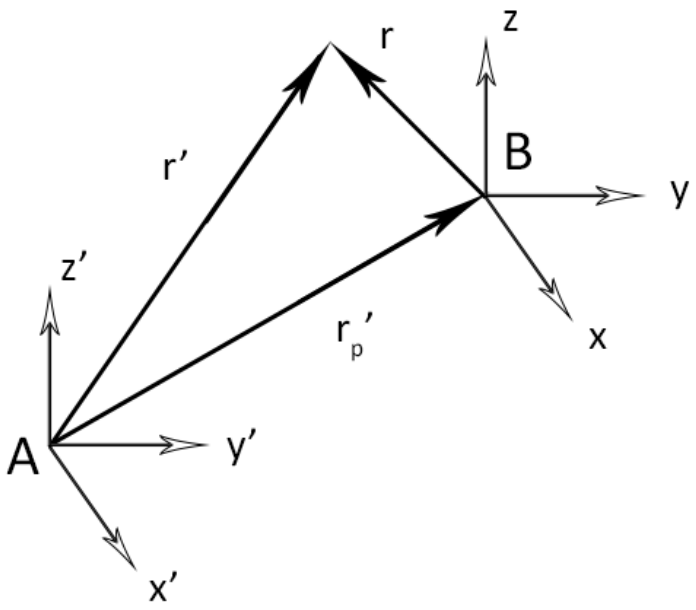


Figure 4.2: Vector Addition

Since $\bar{r}$ is with reference to the body frame, $\int \bar{r}\rho_a dV = 0$

$$\int (\bar{r'} - \bar{r'_p})\rho_a dV = 0$$

$$\int \bar{r'}\rho_a dV = \bar{r'_p} \int \rho_a dV = \bar{r'_p} m$$

$$\bar{r'_p} = \frac{1}{m} \int \bar{r'}\rho_a dV$$

Taking LHS of equation 4.4,

$$\frac{d}{dt} \int_V \rho_a (\frac{d\bar{r'}}{dt}) dV = \frac{d}{dt} \left[\frac{d}{dt} \int_V \rho_a (\bar{r'_p} + \bar{r}) dV \right] = \frac{d}{dt} \left[\frac{d}{dt} (m\bar{r'_p}) \right] = m \frac{d\bar{v_p}}{dt}$$

Taking RHS of equation 4.4,

$$\int_V \rho_a \bar{g} dV + \int_S \bar{F} ds = m\bar{g} + \bar{F}$$

where $\bar{F_x}$ is the net external force acting on the robot

If body frame is rotating with angular velocity w, then any vector $\bar{a}$ in the body frame B can be written in the inertial frame I as following,

$$\left[\frac{d\bar{a}}{dt} \right]_I = \left[\frac{d\bar{a}}{dt} \right]_B + \bar{w} \times \bar{a} \tag{4.6}$$

If $\bar{a}$ is the momentum vector $m\bar{v}$ in the body frame, then,

$$m\left[\frac{d\bar{v}}{dt}\right]_I = m\left[\frac{d\bar{v}}{dt}\right]_B + \bar{w} \times m\bar{v}$$

Rewriting equation 4.4,

$$m\left[\frac{d\bar{v}}{dt}\right]_B + \bar{w} \times m\bar{v} = [m\bar{g}]_B + \left[\bar{F}_x\right]_B \tag{4.7}$$

where

$$[m\bar{g}]_B = \begin{bmatrix} -mgsin\theta \\ mgsin\phi cos\theta \\ mgcos\theta cos\phi \end{bmatrix}$$

Taking LHS of equation 4.5,

$$\frac{d}{dt}\int_V \bar{r} \times \rho_a(\frac{d\bar{r'}}{dt})dV = \int_V \frac{d}{dt}\left[\bar{r} \times \rho_a(\frac{d\bar{r'}}{dt})\right]dV = \int_V \bar{r} \times \frac{d}{dr}(\frac{d\bar{r}}{dt})\rho_a dV$$

From equation 4.6, we can write $\left[\frac{d\bar{r}}{dt}\right]_I = \left[\frac{d\bar{r}}{dt}\right]_B + \bar{w} \times \bar{r}$

But, $\left[\frac{d\bar{r}}{dt}\right]_B = 0$ since the robot is considered as a rigid body.

Hence

$$\frac{d}{dt}\int_V \bar{r} \times \rho_a(\frac{d\bar{r'}}{dt})dV = \int_V \bar{r} \times \frac{d}{dr}(\bar{w} \times \bar{r})\rho_a dV$$

$$= \int_V \bar{r} \times \left[\frac{\partial}{\partial t}(\bar{w} \times \bar{r}) + \bar{w} \times (\bar{w} \times \bar{r})\right]\rho_a dV$$

$$= \int_V \bar{r} \times \left[(\dot{\bar{w}} \times \bar{r}) + \bar{w} \times (\bar{w} \times \bar{r})\right]\rho_a dV$$

Equation 4.5 reduces to

$$\int_V \bar{r} \times \left[(\dot{\bar{w}} \times \bar{r})\right]\rho_a dV + \int_V \bar{r} \times \left[\bar{w} \times (\bar{w} \times \bar{r})\right]\rho_a dV = \bar{M} \tag{4.8}$$

where $\bar{M}$ is the net external torque acting on the robot.

Taking the first term in the LHS of equation 4.8,

$$\int_V \bar{r} \times \left[(\dot{\bar{w}} \times \bar{r})\right]\rho_a dV = \int_V [\dot{\bar{w}}(\bar{r}.\bar{r}) - \bar{r}(\bar{r}.\dot{\bar{w}}]\rho_a dV = \dot{\bar{w}}\int r^2 dm - \int r(r.\dot{\bar{w}})dm$$

$$= \begin{bmatrix} \dot{P} & \dot{Q} & \dot{R} \end{bmatrix}^T \int (x^2 + y^2 + z^2)dm - \begin{bmatrix} x & y & z \end{bmatrix}^T \int (x\dot{P} + y\dot{Q} + z\dot{R})dm$$

$$\int_V \bar{r} \times [(\dot{\bar{w}} \times \bar{r})] \, \rho_a dV = \begin{bmatrix} \int (y^2+z^2)dm & -\int xydm & -\int xzdm \\ -\int yxdm & \int (x^2+z^2)dm & -\int yzdm \\ -\int zxdm & -\int zydm & \int (x^2+y^2)dm \end{bmatrix} \begin{bmatrix} \dot{P} \\ \dot{Q} \\ \dot{R} \end{bmatrix}$$

$$\int_V \bar{r} \times [(\dot{\bar{w}} \times \bar{r})] \, \rho_a dV = \begin{bmatrix} I_{xx} & -I_{xy} & -I_{xz} \\ -I_{xy} & I_{yy} & -I_{yz} \\ -I_{xz} & -I_{yz} & I_{zz} \end{bmatrix} \begin{bmatrix} \dot{P} \\ \dot{Q} \\ \dot{R} \end{bmatrix} \tag{4.9}$$

Taking the second term in the LHS of equation 4.8,

$$\int_V \bar{r} \times [\bar{w} \times (\bar{w} \times \bar{r})] \, \rho_a dV = \int_V \bar{r} \times [\bar{w}(\bar{w}.\bar{r}) - \bar{r}(\bar{w}.\bar{w})] \, \rho_a dV = \int_V (\bar{r} \times \bar{w}(\bar{w}.\bar{r}))\rho_a dV$$

$$= \int \begin{bmatrix} x \\ y \\ z \end{bmatrix} \times \begin{bmatrix} P(Px+Qy+Rz) \\ Q(Px+Qy+Rz) \\ R(Px+Qy+Rz) \end{bmatrix} dm$$

$$= \int \begin{bmatrix} PRxy + QRy^2 + R^2yz - PQxz - Q^2zy - QRz^2 \\ P^2zx + PQyz + PRz^2 - PRx^2 - QRxy - R^2xz \\ PQx^2 + Q^2xy + QRxz - P^2xy - PQy^2 - PRz^2 \end{bmatrix} dm$$

$$= \begin{bmatrix} I_{xy}PR + I_{yz}(R^2-Q^2) - I_{xz}PQ + (I_{zz}-I_{yy})RQ \\ (I_{xx}-I_{zz})PR + I_{xz}(P^2-R^2) - I_{xy}QR + I_{yz}PQ \\ (I_{yy}-I_{xx})PQ + I_{xy}(Q^2-P^2) + I_{xz}QR - I_{yz}PR \end{bmatrix}$$

$$\int_V \bar{r} \times [\bar{w} \times (\bar{w} \times \bar{r})] \, \rho_a dV = \begin{bmatrix} P \\ Q \\ R \end{bmatrix} \times \begin{bmatrix} I_{xx} & -I_{xy} & -I_{xz} \\ -I_{xy} & I_{yy} & -I_{yz} \\ -I_{xz} & -I_{yz} & I_{zz} \end{bmatrix} \begin{bmatrix} P \\ Q \\ R \end{bmatrix} \tag{4.10}$$

Using equations 4.9 and 4.10, equation 4.8 can be written as,

$$[I\dot{\bar{w}}]_B + [\bar{w} \times I\bar{w}]_B = [\bar{M}]_B \tag{4.11}$$

Combining the equations 4.7 and 4.11,

$$\begin{bmatrix} mI_{3X3} & 0_{3X3} \\ 0_{3X3} & I \end{bmatrix} \begin{bmatrix} \dot{v} \\ \dot{w} \end{bmatrix} + \begin{bmatrix} \bar{w} \times m\bar{v} \\ \bar{w} \times I\bar{w} \end{bmatrix} = \begin{bmatrix} \bar{F} \\ \bar{M} \end{bmatrix} \tag{4.12}$$

where I_{3X3} is a unit matrix of size 3

0_{3X3} is a null matrix of size 3

I is the inertia matrix

$$\bar{F} = \begin{bmatrix} F_x & F_y & F_z \end{bmatrix}^T$$
$$\bar{M} = \begin{bmatrix} M_x & M_y & M_z \end{bmatrix}^T$$
All the variables are with respect to the body frame

Expanding equation 4.12,

$$m(\dot{u} - vR + wQ) = -mgsin\theta + F_x$$

$$m(\dot{v} + uR - wP) = mgcos\theta sin\phi + F_y$$

$$m(\dot{w} - uQ + vP) = mgcos\theta cos\phi + F_z$$

$$I_{xx}\dot{P} - I_{xy}\dot{Q} - I_{xz}\dot{R} + I_{xy}PR + I_{yz}(R^2 - Q^2) - I_{xz}PQ + (I_{zz} - I_{yy})RQ = M_x$$

$$-I_{xy}\dot{P} + I_{yy}\dot{Q} - I_{yz}\dot{R} + (I_{xx} - I_{zz})PR + I_{xz}(P^2 - R^2) - I_{xy}QR + I_{yz}PQ = M_y$$

$$-I_{xz}\dot{P} - I_{yz}\dot{Q} + I_{zz}\dot{R} + (I_{yy} - I_{xx})PQ + I_{xy}(Q^2 - P^2) + I_{xz}QR - I_{yz}PR = M_z$$

Assuming the robot does not translate along Z axis and does not rotate about X and Y axis, given the planar configuration of the thrusters,

$$\theta = \phi = w = P = Q = 0, \qquad \dot{w} = \dot{P} = \dot{Q} = 0.$$

Substituting these values in the above equations, we get,

$$m\dot{u} - mvR = F_x$$

$$m\dot{v} + muR = F_y$$

$$-mg = F_z \text{ (Bouyant force)}$$

$$I_{zz}\dot{R} = M_z$$

The above equations can be rewritten as,

$$\dot{u} = vR + \frac{F_x}{m}$$

$$\dot{v} = -uR + \frac{F_y}{m}$$

$$\dot{R} = \frac{M_z}{I_{zz}}$$

These equations can be combined with the kinematic equations and below is the

complete state-space representation of the system,

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\psi} \\ \dot{u} \\ \dot{v} \\ \dot{R} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & cos\psi & -sin\psi & 0 \\ 0 & 0 & 0 & sin\psi & cos\psi & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & R & 0 \\ 0 & 0 & 0 & -R & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ \psi \\ u \\ v \\ R \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ \frac{1}{m} & 0 & 0 \\ 0 & \frac{1}{m} & 0 \\ 0 & 0 & \frac{1}{I_{zz}} \end{bmatrix} \begin{bmatrix} F_x \\ F_y \\ M_z \end{bmatrix} \tag{4.13}$$

$$\begin{bmatrix} x \\ y \\ \psi \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ \psi \\ u \\ v \\ R \end{bmatrix} \tag{4.14}$$

It is observed that equation 4.13 is nonlinear. For the ease of analysis, it can be linearized about an equilibrium point by using Taylor series expansion.
Take

$$\dot{x} = ucos\psi - vsin\psi = f_1$$

$$\dot{y} = usin\psi + vcos\psi = f_2$$

$$\dot{\psi} = R = f_3$$

$$\dot{u} = vR + \frac{F_x}{m} = f_4$$

$$\dot{v} = -uR + \frac{F_y}{m} = f_5$$

$$\dot{R} = \frac{M_z}{I_{zz}} = f_6$$

The linearized form of the system can be written as,

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\psi} \\ \dot{u} \\ \dot{v} \\ \dot{R} \end{bmatrix} = \begin{bmatrix} \frac{\partial f_1}{\partial x} & \frac{\partial f_1}{\partial y} & \frac{\partial f_1}{\partial \psi} & \frac{\partial f_1}{\partial u} & \frac{\partial f_1}{\partial v} & \frac{\partial f_1}{\partial R} \\ \frac{\partial f_2}{\partial x} & \frac{\partial f_2}{\partial y} & \frac{\partial f_2}{\partial \psi} & \frac{\partial f_2}{\partial u} & \frac{\partial f_2}{\partial v} & \frac{\partial f_2}{\partial R} \\ \frac{\partial f_3}{\partial x} & \frac{\partial f_3}{\partial y} & \frac{\partial f_3}{\partial \psi} & \frac{\partial f_3}{\partial u} & \frac{\partial f_3}{\partial v} & \frac{\partial f_3}{\partial R} \\ \frac{\partial f_4}{\partial x} & \frac{\partial f_4}{\partial y} & \frac{\partial f_4}{\partial \psi} & \frac{\partial f_4}{\partial u} & \frac{\partial f_4}{\partial v} & \frac{\partial f_4}{\partial R} \\ \frac{\partial f_5}{\partial x} & \frac{\partial f_5}{\partial y} & \frac{\partial f_5}{\partial \psi} & \frac{\partial f_5}{\partial u} & \frac{\partial f_5}{\partial v} & \frac{\partial f_5}{\partial R} \\ \frac{\partial f_6}{\partial x} & \frac{\partial f_6}{\partial y} & \frac{\partial f_6}{\partial \psi} & \frac{\partial f_6}{\partial u} & \frac{\partial f_6}{\partial v} & \frac{\partial f_6}{\partial R} \end{bmatrix} \begin{bmatrix} x \\ y \\ \psi \\ u \\ v \\ R \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ \frac{1}{m} & 0 & 0 \\ 0 & \frac{1}{m} & 0 \\ 0 & 0 & \frac{1}{I_{zz}} \end{bmatrix} \begin{bmatrix} F_x \\ F_y \\ M_z \end{bmatrix}$$

Substituting f_1, f_2, f_3, f_4, f_5 and f_6, we get

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\psi} \\ \dot{u} \\ \dot{v} \\ \dot{R} \end{bmatrix} = \begin{bmatrix} 0 & 0 & -u_o sin\psi_o - v_o cos\psi_o & cos\psi_o & -sin\psi_o & 0 \\ 0 & 0 & u_o cos\psi_o - v_o sin\psi_o & sin\psi_o & cos\psi_o & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & R_o & v_o \\ 0 & 0 & 0 & -R_o & 0 & -u_o \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ \psi \\ u \\ v \\ R \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ \frac{1}{m} & 0 & 0 \\ 0 & \frac{1}{m} & 0 \\ 0 & 0 & \frac{1}{I_{zz}} \end{bmatrix} \begin{bmatrix} F_x \\ F_y \\ M_z \end{bmatrix}$$

where u_o, v_o, R_o and ψ_o are values of u, v, R and ψ at equilibrium. Assuming $u_o = v_o = R_o = \psi_o = 0$, the above equation reduces to

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\psi} \\ \dot{u} \\ \dot{v} \\ \dot{R} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ \psi \\ u \\ v \\ R \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ \frac{1}{m} & 0 & 0 \\ 0 & \frac{1}{m} & 0 \\ 0 & 0 & \frac{1}{I_{zz}} \end{bmatrix} \begin{bmatrix} F_x \\ F_y \\ M_z \end{bmatrix} \tag{4.15}$$

Equations 4.14 and 4.15 together form the linearized state space representation of the system.

CHAPTER 5

CONTROL OF THE USV

The USV can operate in two configurations - non-holonomic model (or unicyclic) and holonomic model.

Under the non-holonomic configuration, the robot does not have any y-velocity in the body frame. In other words, the linear velocities in the global frame are dependent on the angular orientation. Hence, to translate in y-direction in the inertial frame, the robot must turn. To achieve this, the lateral thrusters in the front and the rear are disabled.

A holonomic drive is achieved when the controllable degrees of freedom are at least equal to the total degrees of freedom. The designed USV is an over-actuated system as the controllable degrees of freedom are four - one for each thruster. Assuming that the motion of the USV is only on 2D surface of the water, considering that the thrusters are planar, the total degrees of freedom are three. Hence, the designed robot is capable of holonomic motion, in which the each control parameter is independent of other parameters.

5.1 Plant Model

Equation 4.14 and 4.15 can be written individually for x, y, and ψ to get a Single Input Single Output (SISO) Linear Time Invariant (LTI) causal form ($\dot{X} = AX + BU$, $Y = C \cdot X + DU$, where D = [0] for our case), as

$$\begin{bmatrix} \dot{x} \\ \dot{u} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ u \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix} \cdot \begin{bmatrix} 0 \\ F_x \end{bmatrix}, \begin{bmatrix} x \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \cdot \begin{bmatrix} x \\ u \end{bmatrix}$$

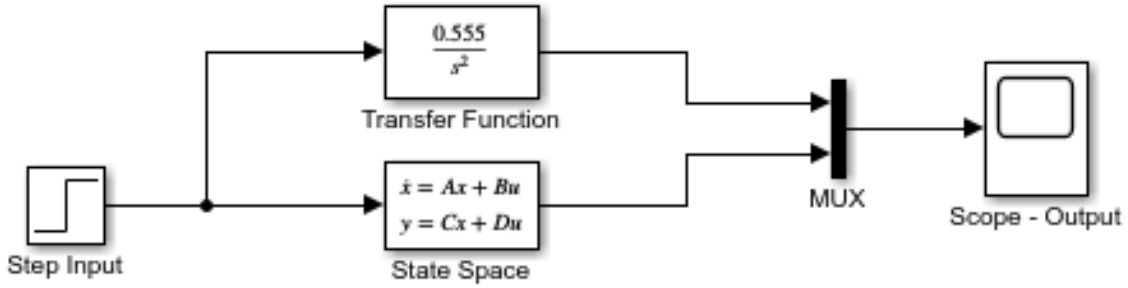
$$\begin{bmatrix} \dot{y} \\ \dot{v} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} y \\ v \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix} \cdot \begin{bmatrix} 0 \\ F_y \end{bmatrix}, \begin{bmatrix} y \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \cdot \begin{bmatrix} y \\ v \end{bmatrix}$$

$$\begin{bmatrix} \dot{\psi} \\ \dot{R} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \psi \\ R \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{I_{zz}} \end{bmatrix} \cdot \begin{bmatrix} 0 \\ M_z \end{bmatrix}, \begin{bmatrix} \psi \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \cdot \begin{bmatrix} \psi \\ R \end{bmatrix}$$

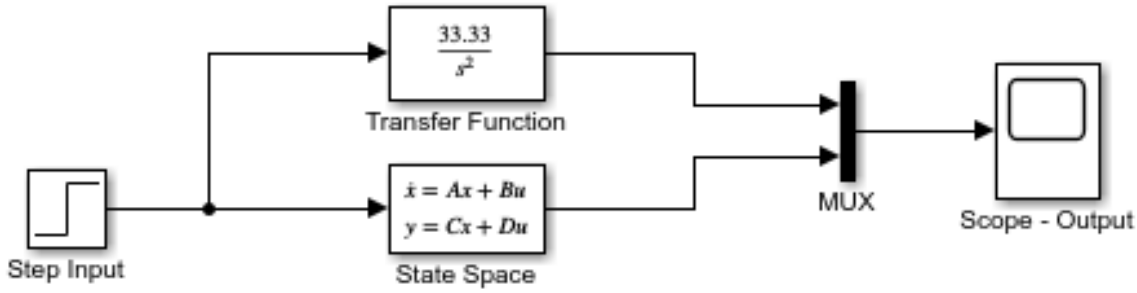
These three SISO LTI forms can be converted to get the transfer function of the individual plants. Substituting the values of m and I_{zz}, the transfer functions for x, y and ψ are respectively,

$$Tf_x = \frac{0.555}{s^2}, \qquad Tf_y = \frac{0.555}{s^2}, \qquad Tf_\psi = \frac{33.33}{s^2}$$

Figures 5.2 (a) and 5.2 (b) depict the transfer functions $\frac{0.555}{s^2}$ and $\frac{33.33}{s^2}$ from the plants shown in figure 5.1.

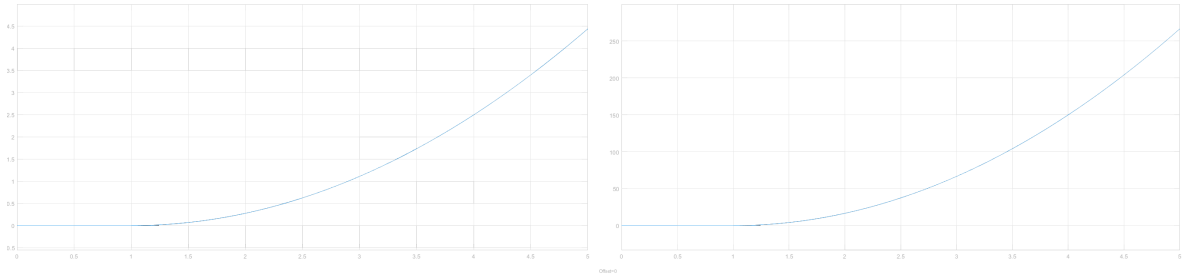


(a) Linear Velocities Plant



(b) Angular Velocity Plant

Figure 5.1: Plant or Process



(a) Plant response for Linear Velocities (b) Plant response for Angular Velocities

Figure 5.2: Plots of transfer functions with time

5.2 PID Controller

A PID control algorithm is used for achieving the go to goal of the robot. As PID algorithm provides a simple yet effective control in most scenarios, this was used for our application. PID stands for Proportional-Integral-Derivative which means, the command signal is a linear combination of these three operators over error. Each of the error terms has a corresponding gain which shall be tuned to improve the response.

5.2.1 Proportional Controller

In this case, the command signal is proportional to the difference between the set point and the output, which is the error signal. Thus, when the error is large, the command input is larger and thus, it quickly approaches the set point. When the error reduces, the command signal strength is low causing the response to be slow near the set point. In certain cases, a steady state error may prevail and thus, only proportional control may not be enough.

5.2.2 Proportional-Integral (PI) Controller

A steady state error usually persists in case of P controller. To mitigate the zero or offset error, an integral term is used. This term sums up the error over the period of time until the error reaches zero and thus, the cumulative error develops the action potential to diminish the offset.

5.2.3 Proportional-Integral-Derivative (PID) Controller

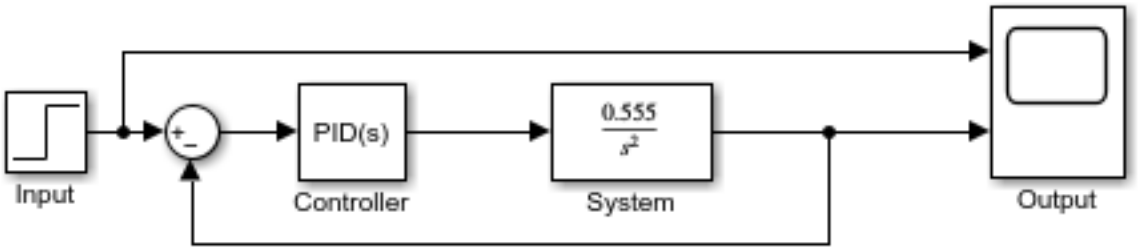
A PI controller may not give the best results as it cannot predict the future behaviour of the error, i.e. it is a causal system. Thus, when the error is large in case of P or PI controller, the command signal may cause the system to overshoot. Introducing a Derivative or D term in the control helps to predict the future variations of the error. The error's rate of change is determined and a command signal proportional to all the three operations is provided to the system. This minimizes the peak overshoot, decreases the rise time and steady state error. The gains can be tuned to optimality to get the best response in shortest time interval. Table 5.1 shows the system response to change in the PID gains or constants.

K	Rise Time	Overshoot	Settling Time	Steady-State Error	Stability
K_p	Decrease	Increase	Negligible	Decrease	Decrease
K_i	Decrease	Increase	Increase	Decrease	Decrease
K_d	Negligible	Decrease	Decrease	No change	Increase

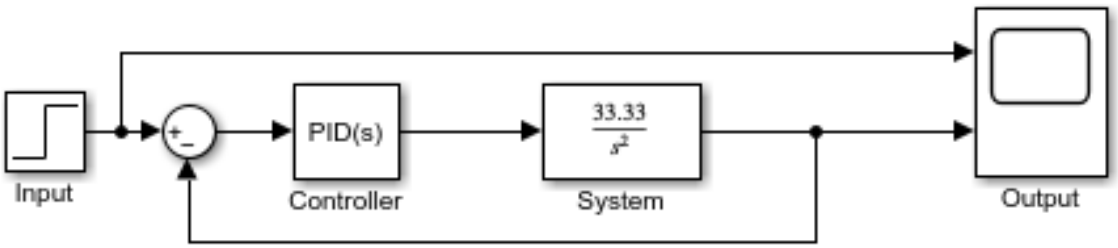
Table 5.1: System response to increase in PID Gains [15]

5.3 Implementation

A PID controller is implemented in each of the plants shown in Fig. 5.1. The block diagrams of the resulting control systems with feedback are shown in Fig. 5.3. The PID gains are tuned to get a stable response and the corresponding step responses are shown in Fig. 5.4 and ramp responses in Fig. 5.5.

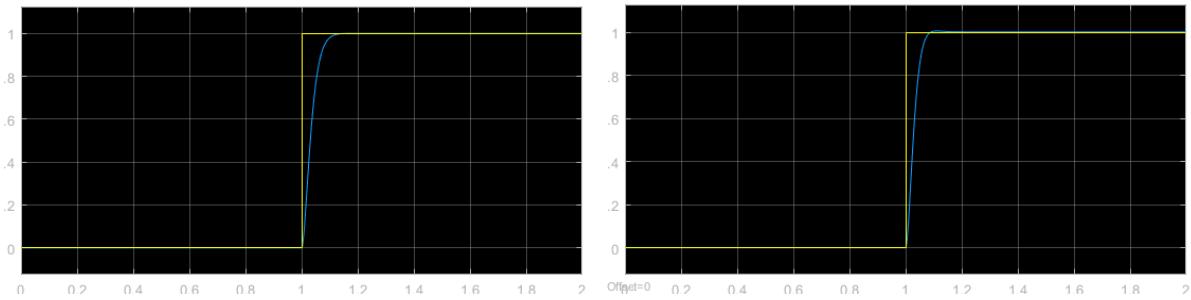


(a) Linear Velocities System



(b) Angular Velocity System

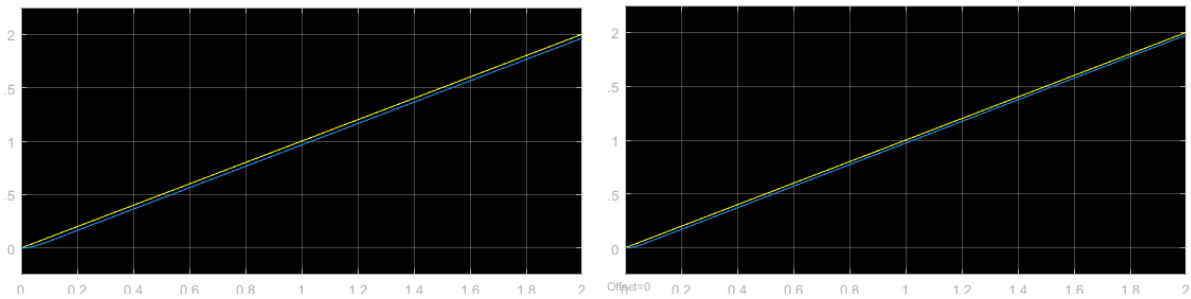
Figure 5.3: System with PID Controller



(a) Controller for Linear Velocities

(b) Controller for Angular Velocity

Figure 5.4: Step Response of the Controllers (Amplitude vs Time plot)



(a) Controller for Linear Velocity

(b) Controller for Angular Velocity

Figure 5.5: Ramp Response of the Controllers (Amplitude vs Time plot)

CHAPTER 6

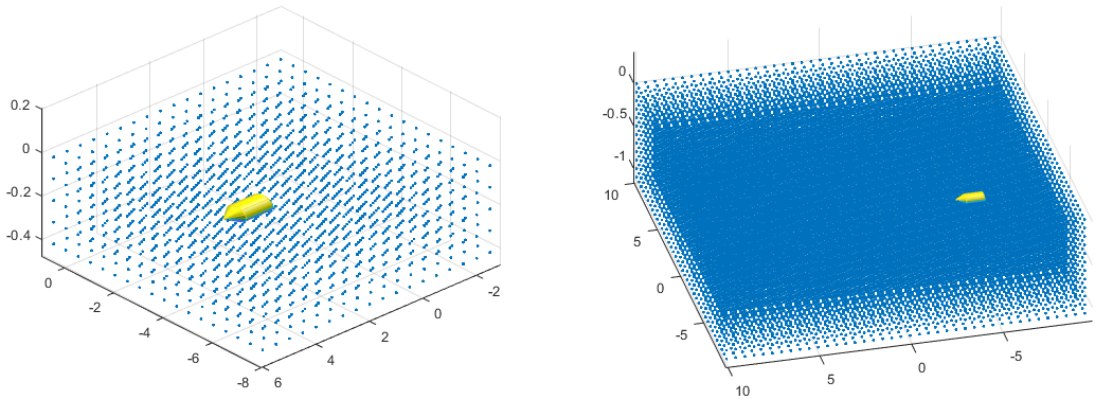
RESULTS AND CONCLUSION

The present section is devoted to the presentation of response of the Unmanned Surface Vehicle (USV) to a given goal position or trajectory.

6.1 Simulation Model

A simulation model has been implemented in real-time using MATLAB's surface, hgtransform, meshgrid and opengl functions and packages. The water is represented by creating a meshgrid in x, y and z directions for a specific range in each axis and then plotted using a scatter plot. Two light sources with a specific ray direction from infinity were positioned at two opposite ends. The USV model is created using the cylinder function, which generates points. A cylinder makes the body and two cones make the front and the rear of the robot. These individual set of points are made into surface using the surface function. These surfaces are linked to each other to make a complete model using hgtransform. Finally, the setup is rendered using opengl. The position and orientation of the robot is altered every instant using the corresponding transformation matrices. Figure 6.1 depicts the simulation in a three-dimensional plot.

The maximum thrusts of each of the USV's thrusters have been limited to 10N and consequently, the maximum net force along each axis is limited to 20N moment is limited to 3Nm. The simulation time is set for 35s with a sampling time of 0.03s.



(a) (b)

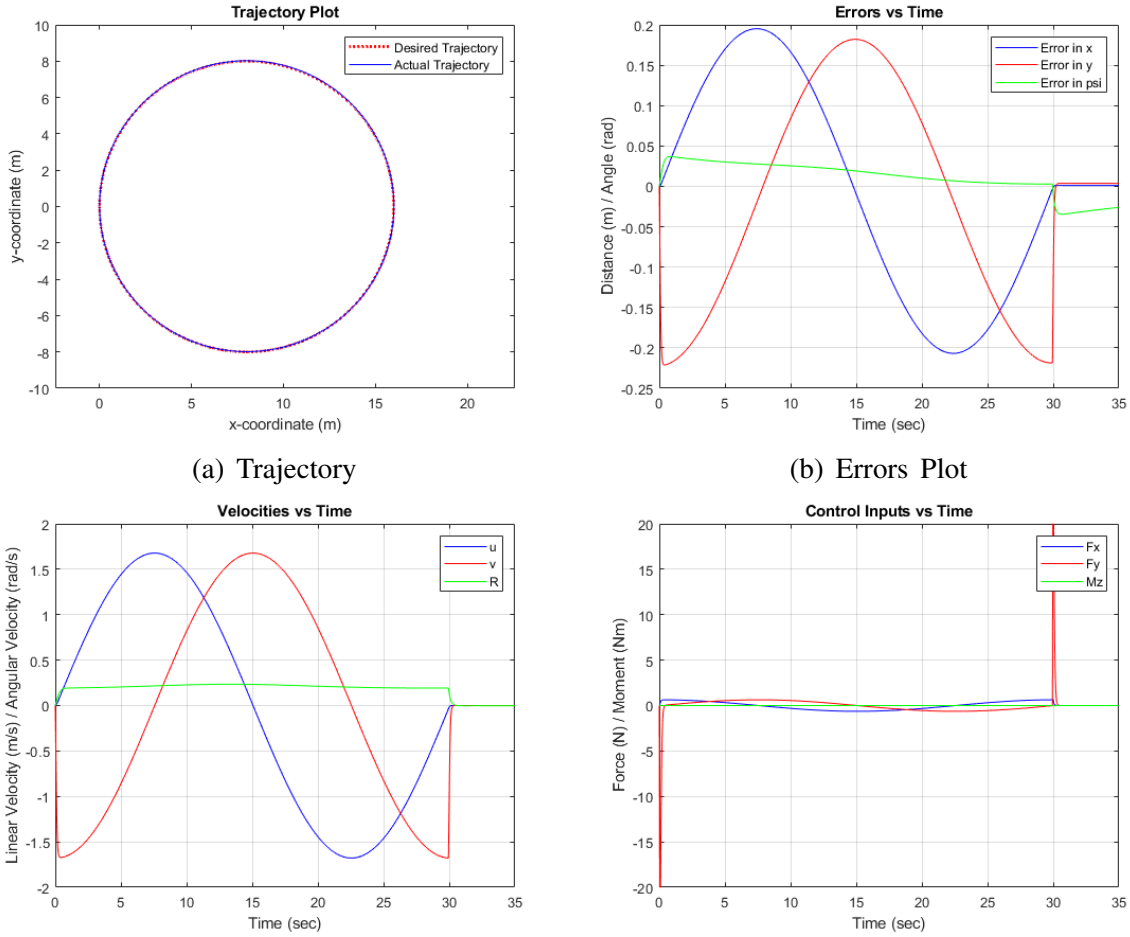
Figure 6.1: Simulation Model

6.2 Response of the USV to Different Trajectories

A set of three trajectories - circle, square and sine curves were taken to test the performance of the USVs.

6.2.1 Circle

Figure 6.2 shows the trajectory with corresponding state errors, velocities and control inputs with time.



(a) Trajectory

(b) Errors Plot

(c) Velocities Plot

(d) Control Inputs Plot

Figure 6.2: Circular Trajectory Response

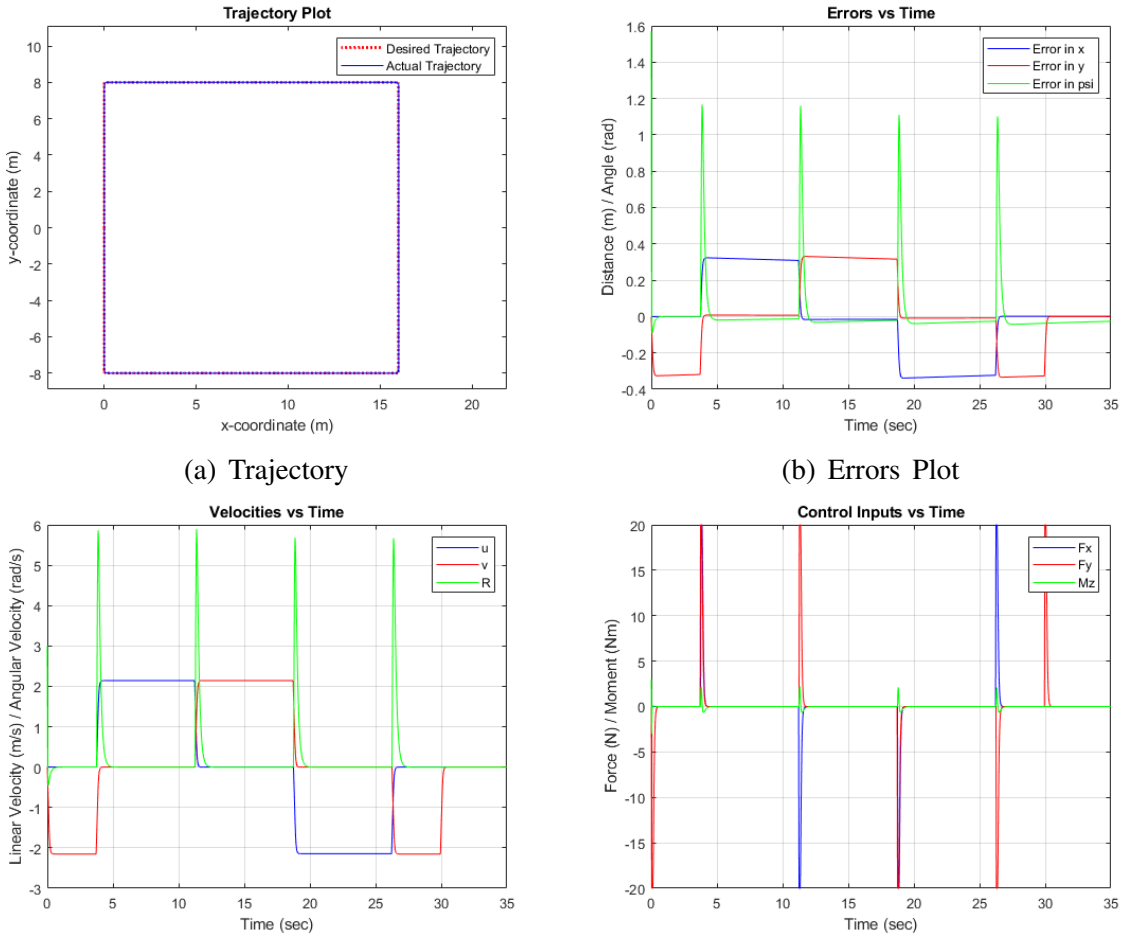
Table 6.1 presents the statistics of various physical parameters during the course of motion in the circular trajectory. The state error in x ranges from -0.207 to 0.195 m, y from -0.221 to 0.182 m, and ψ from -0.035 to 0.037 rad.

	x error	y error	ψ error	u	v	R	F_x	F_y	M_z
Max	0.195	0.182	0.037	1.678	1.678	0.235	0.712	20	0.016
Min	-0.207	-0.221	-0.035	-1.678	-1.678	-0.002	-0.633	-20	-0.029
Avg	-0.005	-0.016	0.011	0.0	0.001	0.18	0.0	0.0	0.0
Var	0.017	0.017	0.0	1.207	1.199	0.006	0.171	2.91	0.0
SD	0.132	0.132	0.02	1.099	1.095	0.075	0.414	1.706	0.002

Table 6.1: Parameters for Circular Trajectory (in SI units)

6.2.2 Square

Figure 6.3 shows the trajectory with corresponding state errors, velocities and control inputs with time.



(a) Trajectory

(b) Errors Plot

(c) Velocities Plot

(d) Control Inputs Plot

Figure 6.3: Square Trajectory Response

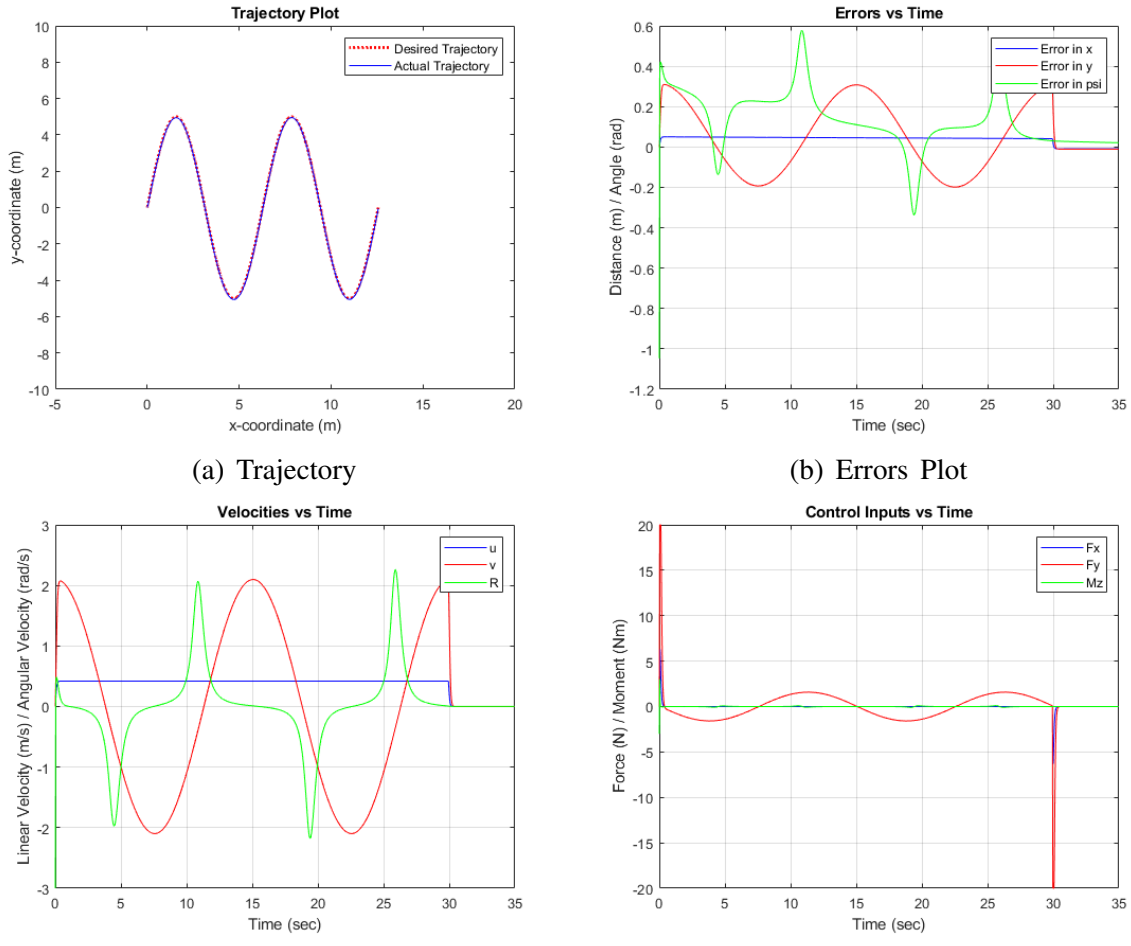
Table 6.2 presents the statistics of various physical parameters during the course of motion in the square trajectory. The state error in x ranges from -0.338 to 0.324 m, y from -0.334 to 0.331 m, and ψ from -0.090 to 1,571 rad.

	x error	y error	ψ error	u	v	R	F_x	F_y	M_z
Max	0.324	0.331	1.571	2.144	2.144	5.898	20	20	3
Min	-0.338	-0.334	-0.090	-2.144	-2.153	-0.452	-20	-20	-3
Avg	-0.006	0.0	0.012	0.0	0.002	0.180	0.0	0.0	0.0
Var	0.044	0.044	0.028	1.942	1.932	0.660	7.340	11.045	0.058
SD	0.211	0.211	0.168	1.394	1.390	0.813	2.709	3.323	0.240

Table 6.2: Parameters for Square Trajectory (in SI units)

6.2.3 Sine Curve

Figure 6.4 shows the trajectory with corresponding state errors, velocities and control inputs with time.



(a) Trajectory

(b) Errors Plot

(c) Velocities Plot

(d) Control Inputs Plot

Figure 6.4: Sine Curve Trajectory Response

Table 6.3 presents the statistics of various physical parameters during the course of motion in the sine curve trajectory. The state error in x ranges from -0.009 to 0.050 m, y from -0.199 to 0.310 m, and ψ from -1.047 to 0.576 rad.

	x error	y error	ψ error	u	v	R	F_x	F_y	M_z
Max	0.050	0.310	0.576	0.420	2.098	2.260	6.292	20	3
Min	-0.009	-0.199	-1.047	0.0	-2.097	-3	-6.292	-20	-3
Avg	0.038	0.046	0.130	0.359	-0.002	0.011	0.0	0.0	0.0
Var	0.0	0.027	0.021	0.022	1.871	0.398	0.201	4.646	0.016
SD	0.019	0.166	0.144	0.147	1.368	0.631	0.449	2.155	0.128

Table 6.3: Parameters for Sine Curve Trajectory (in SI units)

6.3 Conclusion

An effective USV design for oil spill cleanup is proposed and a 3D CAD model is made in SolidWorks 2016. The USV is mathematically modelled and PID control is implemented. The response of the USV is studied for different trajectories and the variation of several parameters with time is plotted. A smooth response with a significantly low error can be observed for curved and straight trajectories. In the case of square trajectory, the sharp turns result in a sudden spike of the error values, velocities and control inputs at the corners. At such instances, actuator saturation is avoided by enabling the PID controller to work in a restricted range of control inputs. The USV was able to trace the square trajectory accurately with a decent response. The overall performance of the USV is sufficiently accurate to be implemented practically.

6.4 Future Work

The design and control of the USV presented in this study promises to be a viable alternative for the conventional oil cleanup techniques. The USV is capable of holonomic motion, which enables easy cleaning up of oil from any direction. In future editions, the design could incorporate a hydraulic mechanism to enable inclination of thrusters to adapt to waves. The dynamic model of the system is close to real world and could be extended by including drag forces and wave models to accurately replicate the actual scenario. The control algorithm works efficiently for almost every kind of motion including sharp turns or gradual curves. Modern and advanced controllers like Adaptive and Model Predictive Control could be implemented. The next step to implement autonomous path planning like Dijkstra's or A* algorithms, and guidance to cover a particular patch area. Further, a swarm intelligence can be implemented on multiple robots of this kind sto effectively clean the spill in a shorter span of time. Also, reinforcement learning techniques could be implemented as an alternative to these control and path planning techniques.

APPENDIX A

Code Attachments

GitHub Repository: https://github.com/gnitish18/USV-Oil_Spill [16]

A.1 Main Program

```
%% PID Control of an Unmanned Surface Vehicle (USV) using a dynamic model
clear; close all; clc;

sim_time = 100;
sampling = 0.1;

%% Set-up Simulation Model

figure(1)
ax = axes('XLim',[-10 10],'YLim',[-5 20],'ZLim',[-2.5 2.5]);
view(3);    % view all the axes (3D)
grid on;

% Modelling water
Water_x=-10:0.5:10;
Water_y=-10:0.5:20;
Water_z=-1.5:0.1:1;
[Water_X,Water_Y,Water_Z] = meshgrid(Water_x,Water_y,Water_z);
I = Water_Z <= 0;
scatter3(Water_X(I),Water_Y(I),Water_Z(I),'.');

% Light source
light('Position',[-10 -10 2],'Style','infinite')
light('Position',[10 10 2.5],'Style','infinite')

% Robot (USV) Model
[con_x,con_y,con_z] = cylinder([0.1 0.0],12);
[cyl_x,cyl_y,cyl_z] = cylinder([0.2 0.2],12);
% Surfaces of USV
h(1) = surface(4*con_x,0.5*con_z,-0.3*con_y,'FaceColor','y');
h(2) = surface(2*cyl_x,-1*cyl_z,0.15*cyl_y,'FaceColor','y');
h(3) = surface(4*con_x,(0.01*con_z)-1,-0.3*con_y,'FaceColor','y');
h(4) = surface(0.1,1,0.1,'FaceColor','w');
h(1).EdgeColor = 'none'; h(1).FaceLighting = 'gouraud';
h(2).EdgeColor = 'none'; h(2).FaceLighting = 'gouraud';
```

```
h(3).EdgeColor = 'none'; h(3).FaceLighting = 'gouraud';
% Transforming the surfaces wrt robot position
t = hgtransform('Parent',ax);
set(h,'Parent',t)
set(gcf,'Renderer','opengl')
drawnow

%% Trajectories, Control and Guidance of USV

% Trajectory selection
disp('Enter trajectory:');
traj = input(' 1. Circle \n 2. Square \n 3. Sine \n');
switch traj
    % Circle
    case 1
        pose = [0;0;-pi/2;0;0;0];
        n = sim_time/sampling; R = 8;
        k = 2*pi*linspace(0,1,n)' - 3*pi*ones(n,1);
        x = R*ones(n,1) + R.*cos(k);
        y = zeros(n,1) + R.*sin(k);
    % Square
    case 2
        pose = [0;0;-pi/2;0;0;0];
        n = sim_time/sampling; n1 = fix(n/4);
        x0 = zeros((n1/2),1);
        y0 = linspace(0,-8,fix(n1/2))';
        x1 = linspace(0,16,n1)';
        y1 = -8*ones(n1,1);
        x2 = 16*ones(n1,1);
        y2 = linspace(-8,8,n1)';
        x3 = linspace(16,0,n1)';
        y3 = 8*ones(n1,1);
        x4 = zeros((n1/2),1);
        y4 = linspace(8,0,fix(n1/2))';
        x = [x0;x1;x2;x3;x4];
        y = [y0;y1;y2;y3;y4];
    % Sine
    case 3
        pose = [0;0;pi/3;0;0;0];
        n = sim_time/sampling;
        x = 4*pi*linspace(0,1,n)';
        y = 5*sin(x);
end

goal = [x, y];
[pose] = Controller(goal, pose, t);
```

A.2 Control Algorithm

```
%% Go to goal using Holonomic Dynamic Model of the system
function [pose] = Controller(goal, pose, t)

    % Sampling rate for simulation
    sample_time = 0.03;
    sim_time = 35;
    vizRate = rateControl(1/sample_time);
    timestep = 0:sample_time:sim_time;

    % Limits of thrusters
    Max_Thrust = 20;
    Max_Moment = 3;

    % PID gains
    Kp = [0.1;0.1;0.01];
    Kd = [500;500;5];
    Ki = [0;0;0];

    % Mass and Moment of Inertia
    m = 1.8;
    I_zz = 0.03;

    % State-Space equation: x_dot = A*x + B*u
    %   Where, x is state vector, u is control variables vector
    %          A is state matrix, B is input matrix

    A = [0, 0, 0, 1, 0, 0;
         0, 0, 0, 0, 1, 0;
         0, 0, 0, 0, 0, 1;
         0, 0, 0, 0, 0, 0;
         0, 0, 0, 0, 0, 0;
         0, 0, 0, 0, 0, 0];

    B = [0    0    0;
         0    0    0;
         0    0    0;
         1/m  0    0;
         0    1/m  0;
         0    0    1/I_zz;];

    % Output equation: y = C*x + D*u
    %   Where, x is state vector, u is control variables vector
    %          C is output matrix, D is direct transmission term (0 here)

```

```
45      C = [1, 0, 0, 0, 0, 0;
46           0, 1, 0, 0, 0, 0;
47           0, 0, 1, 0, 0, 0;];
48
49      % Error variables
50      E = 0;
51      e_old = 0;
52
53      % Lists for plotting data
54      x_list = zeros(fix(sim_time/sample_time),1);
55      y_list = zeros(fix(sim_time/sample_time),1);
56      x_e_list = zeros(fix(sim_time/sample_time),1);
57      y_e_list = zeros(fix(sim_time/sample_time),1);
58      psi_e_list = zeros(fix(sim_time/sample_time),1);
59      u_list = zeros(fix(sim_time/sample_time),1);
60      v_list = zeros(fix(sim_time/sample_time),1);
61      r_list = zeros(fix(sim_time/sample_time),1);
62      Fx_list = zeros(fix(sim_time/sample_time),1);
63      Fy_list = zeros(fix(sim_time/sample_time),1);
64      Mz_list = zeros(fix(sim_time/sample_time),1);
65
66      % Plot of desired trajectory
67      figure(2)
68      plot(goal(:,1), goal(:,2),':r','LineWidth',2);
69      axis([-5 20 -10 10])
70      hold on;
71
72      % Iterate for the given simulation time
73      for time = 1:length(timestep)
74
75          % Update goal points
76          if time < length(goal)
77              y_d = [goal(time,1);
78                     goal(time,2);
79                     atan2(goal(time,2) - pose(2), goal(time,1) - pose(1))];
80          end
81
82          % Computing the errors in state and updation
83          e = y_d - C*pose;
84          e(3) = atan2(sin(e(3)), cos(e(3)));
85          e_dot = e - e_old;
86          E = E + e;
87          U = Kp.*e + Kd.*e_dot + Ki.*E;
88
89          % Limiting the Thrusts and Moment
90          if U(1) > Max_Thrust
91              U(1) = Max_Thrust;
```

```
92          elseif U(1) < -Max_Thrust
93              U(1) = -Max_Thrust;
94          end
95          if U(2) > Max_Thrust
96              U(2) = Max_Thrust;
97          elseif U(2) < -Max_Thrust
98              U(2) = -Max_Thrust;
99          end
100         if U(3) > Max_Moment
101             U(3) = Max_Moment;
102         elseif U(3) < -Max_Moment
103             U(3) = -Max_Moment;
104         end
105
106         % Update errors
107         e_old = e;
108
109         % Update the state variables
110         pose = pose + (A*pose + B*U)*sample_time;
111
112         % Append the updated data to lists
113         x_list(time) = pose(1);
114         y_list(time) = pose(2);
115         x_e_list(time) = e(1);
116         y_e_list(time) = e(2);
117         psi_e_list(time) = e(3);
118         u_list(time) = pose(4);
119         v_list(time) = pose(5);
120         r_list(time) = pose(6);
121         Fx_list(time) = U(1);
122         Fy_list(time) = U(2);
123         Mz_list(time) = U(3);
124
125         % Plotting the live location of USV
126         figure(2)
127         plot(x_list(1:time),y_list(1:time),'-b')
128         hold on
129
130         % Simulation
131         Simulation(pose, t);
132
133         % Sampling Rate
134         waitfor(vizRate);
135     end
136
137     % Trajectory Plot
138     figure(2)
```

```
139     legend('Desired Trajectory','Actual Trajectory')
140     title('Trajectory Plot')
141     xlabel('x-coordinate (m)')
142     ylabel('y-coordinate (m)')
143
144     % Errors plot
145     figure(3)
146     title('Errors vs Time')
147     xlabel('Time (sec)')
148     ylabel('Distance (m) / Angle (rad)')
149     plot(timestep(1:time),x_e_list(1:time),'-b')
150     hold on
151     plot(timestep(1:time),y_e_list(1:time),'-r')
152     hold on
153     plot(timestep(1:time),psi_e_list(1:time),'-g')
154     hold on
155     legend('Error in x','Error in y','Error in psi')
156     grid on
157     title('Errors vs Time')
158     xlabel('Time (sec)')
159     ylabel('Distance (m) / Angle (rad)')
160
161     % Velocities plot
162     figure(4)
163     title('Velocities vs Time')
164     xlabel('Time (sec)')
165     ylabel('Linear Velocity (m/s) / Angular Velocity (rad/s)')
166     plot(timestep(1:time),u_list(1:time),'-b')
167     hold on
168     plot(timestep(1:time),v_list(1:time),'-r')
169     hold on
170     plot(timestep(1:time),r_list(1:time),'-g')
171     hold on
172     legend('u','v','R')
173     grid on
174     title('Velocities vs Time')
175     xlabel('Time (sec)')
176     ylabel('Linear Velocity (m/s) / Angular Velocity (rad/s)')
177
178     % Control Inputs plot
179     figure(5)
180     title('Control Inputs vs Time')
181     xlabel('Time (sec)')
182     ylabel('Force (N) / Moment (Nm)')
183     plot(timestep(1:time),Fx_list(1:time),'-b')
184     hold on
185     plot(timestep(1:time),Fy_list(1:time),'-r')
```

```
186     hold on
187     plot(timestep(1:time),Mz_list(1:time),'-g')
188     hold on
189     legend('Fx','Fy','Mz')
190     grid on
191     title('Control Inputs vs Time')
192     xlabel('Time (sec)')
193     ylabel('Force (N) / Moment (Nm)')
194 
195     % End of Simulation
196     disp('Simulation Complete')
197 
198 end
```

A.3 Simulation Model

```
1  %% Function to update the robot (USV) position in the Simulation model
2  function Simulation(pose, t)
3      % Translation
4      trans = makehgtform('translate', [pose(2) pose(1) 0]);
5      % Rotation
6      rtz = makehgtform('zrotate',-1*pose(3));
7      % Set new position after transformation
8      set(t,'Matrix',trans*rtz);
9      drawnow
10 end
```

References

[1] Josefina Méndez Eduardo Pásaro and Blanca Laffon. Aguilera Francisco. “Review on the Effects of Exposure to Spilled Oils on Human Health”. In: *Journal of Applied Toxicology 30(4)* (2010), : 291–301.

[2] T H D Sydenstricker Annunciado T R and S C Amico. “Experimental Investigation of Various Vegetable Fibers as Sorbent Materials for Oil Spills.” In: *Marine pollution bulletin 50(11)* (2005).

[3] Johanna Aurell and Brian K Gullett. “Aerostat Sampling of PCDD/PCDF Emissions from the Gulf Oil Spill in Situ Burns.” In: *Environmental science technology 44(24)* (2010).

[4] Milind V Joshi Banerjee Shashwat S and Radha V Jayaram. “Treatment of Oil Spill by Sorption Technique Using Fatty Acid Grafted Sawdust.” In: *Chemosphere 64(6)* (2012).

[5] Duong Nam Nguyen Anh Tuan Hoang Van Viet Pham. “A Report of Oil Spill Recovery Technologies.” In: *International Journal of Applied Engineering Research ISSN 0973-4562 Volume 13* (2018).

[6] Gerald et al. Deschamps. “Oil Removal from Water by Selective Sorption on Hydrophobic Cotton Fibers. 1. Study of Sorption Properties and Comparison with Other Cotton Fiber-Based Sorbents.” In: *Environmental Science & Technology 37(5)* (2003).

[7] Petrissa Eckle, Peter Burgherr, and Edouard Michaux. “Risk of Large Oil Spills: A Statistical Analysis in the Aftermath of Deepwater Horizon”. In: *Environmental Science & Technology 46(23)* (2012).

[8] Mervin. Fingas. “Oil Spill Science and Technology.” In: *Gulf professional publishing.* (2016).

[9] T.V. Prasad Emaad Mohamed H. Zahugi Mohamed M. Shanta. “Oil Spill Cleaning Up Using Swarm of Robots.” In: *Advances in intelligent systems and computing* (2013).

[10] *SeaSwarm*. URL: `http://senseable.mit.edu/seaswarm/index.html`.

[11] Duong Nam Nguyen Anh Tuan Hoang Van Viet Pham. “A Report of Oil Spill Recovery Technologies”. In: *International Journal of Applied Engineering Research ISSN* (2018).

[12] Cenk Sakar Ali cemal Toz Burak Koseoglu. “MARINE ENVIRONMENT PROTECTION: NEW TECHNOLOGIES ON OIL SPILL RESPONSE INDUSTRY”. In: *1st International Congress on Ship and Marine Technology* (2016).

[13] Jianhua Wang, Wei Gu, and Jianxin Zhu. “Design of an Autonomous Surface Vehicle Used for Marine Environment Monitoring”. In: *2009 International Conference on Advanced Computer Control* (2009), : 405–409.

[14] J.R. Davis and ASM International. Handbook Committee. *Metals Handbook Desk Edition 2nd Edition*. 75th anniversary ASM handbooks. Taylor & Francis, 1998. ISBN: 9780871706546. URL: https://books.google.co.in/books?id=IpEnvBtSfPQC.

[15] Kiam Heong Ang, G. Chong, and Yun Li. “PID control system analysis, design, and technology”. In: *IEEE Transactions on Control Systems Technology* 13.4 (2005).

[16] *GitHub Repository*. URL: https://github.com/gnitish18/USV-Oil_Spill.