NTNU
Faculty of Information Technology
and Electrical Engineering

Norwegian University of Science and Technology

Autonomous Docking and Departure with an Unmanned Surface Vehicle (USV)

Legge til og fra kai med en ubemannet overflatefarkost (USV)

TTK4551 - Engineering Cybernetics,
Specialization Project (7.5 Credits)

Simen Rogn Aune

June 3, 2019

Preface

This project is given by Norwegian Defence Research Establishment (FFI) through Norwegian University of Science and Technology (NTNU) as a part of the 2 years masters degree program Cybernetics and Robotics education course. The specialization projects constitute 7.5 credits for a 2 years masters degree student. The master thesis written the following semester will have the same problem as the specialization project, which constitutes to the final 30 credits required to be able to receive the master's degree.

Abstract

The usage areas of autonomy have grown in the last decade. The potential benefit of autonomy in shipping is huge. Whether it is large container vessels that are able to carry more and greener; or a small Unmanned Surface Vehicle (USV) on a mission.

An autonomous USV used in demining can remove the event of damage inflicted on the crew on board. The need for a robust control system is needed, such that we are sure that the vessel operates as indented. If having an autonomous vessel at sea that we have no control over there is no point in implementing it.

It is presented three different concepts when dealing with this problem of autonomous docking and departure. The concepts are based on various theories that can fulfill the necessary requirements to complete the task. The concepts will be tried created and implemented as a part of the master's thesis. The concepts consist of three elements: path-planning, actuator control and detection of the vessel at the pier. Theories that can solve the three elements are presented as a part of the concept.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

In this chapter, an explanation around the task in hand is presented. An explanation about the task in its whole, some background information regarding the task and a motivated picture on why it is a problem wished to be solved will some of the topics in the coming sections. A detailed description of the project will follow, explained the different aspects of the task in hand. It is important to review prior literature at the beginning of a project of this magnitude. It can be helpful to look at similar cases involving each element of the problem. There is no need to repeat work if the work is already done.

There may be needed to make assumptions to any solution, and in that case, a list of assumptions around this project will also be talked about. A background of my field of study, and in which category this project falls. Lastly, an outline of the project report in its entirety will be explained.

1.1 Background

The background of the author of specialization report originates from the cybernetics and robotics master study program at the NTNU. This field of study gives a range of lessons from control systems theory to various courses in modeling, simulation, optimization and algorithms and data structures. A description of this project's task will later in this chapter be presented, and all of the tasks fall nicely into the category of what we have learned prior in the years of studying.

1.2 Motivation

This report is written on behalf of both NTNU and FFI. It is FFI who have sent the problem description to the university for a student to take part in it. FFI have developed their own USV called Odin, which is on its way on becoming fully autonomous. The only element currently missing is autonomously docking and departure of the USV. It is wishful to have a fully automated vessel, where it is planned to use the vessel as a deminer at sea. A fully automated USV means that there is no need for any personnel physically on the boat at the time where there may be any danger concerning mine detonating. Safety is a huge factor.

1.3 Project Description

It can generally be a challenging task to dock even for the experienced navigator of a vessel. To be able to dock autonomously one should be aware of the vessels maneuverability and the surrounding environment around the craft. For autonomous docking, it is necessary to plan the maneuvers of the vessel. This task is to investigate, develop and implement concepts to autonomously dock and depart a pier. FFIs Odin is equipped with a rotating LiDAR that precisely measure the distance to any object and land area in the vicinity of the craft. Based on the map with distance measurements, the student will be tasked with exploring the necessary control and planning-algorithm required to dock and depart a pier, autonomously. The most central parts of the assignment will be:

- Plan a safe path form a position of USV in the surroundings of the pier to the docking spot and vice versa for departing the pier.
- Usage of control theory in the last phase of the docking procedure
- State estimation for the vessel's contact with the pier

A list of working operations was given:

1. Do a literature search and describe the short literature that is necessary to complete the assignment
2. Develop and/or implement methods to solve the problem
3. Conduct simulations and physical tests on FFIs USV Odin
4. Summarize the results on the assignment, compare your findings with what has been done in the literature and point out the way for future work

The title of this project is `Autonomous Docking and Departing With an Unmanned Surface Vehicle (USV)`. The objective is to create algorithms and controllers that takes care of the vessels maneuver in the case of autonomously docking and departing a pier. Prior to the start of this project, the USV developed by FFI is almost fully automated. It is only the docking part (and departing) that is missing in the sense of making the USV depart the pier, navigating at sea, and finally back to the docking spot, autonomously. FFI have developed control- and path-planning algorithms that make the USV operate autonomously at a distance from land. Sensors on the USV create a map of the surrounding area that identifies an object in near distance of the vessel. FFI have also implemented collision avoidance to the USV. If there are any objects detected that intervene with the planned path, the vessel will navigate around it. A new path is calculated. Also, if another boat is detected, the USV will follow the International Regulations for Preventing Collisions at Sea (COLREGs)

There is a lot of useful techniques and strategies already developed and tested that can be transferred to the task in hand. The concepts introduced at a later stage of the report can take advantage of what FFI already have created.

1.3.1 FFIs Odin

The USV created by FFI is called Odin. It is thought that the USV one day will operate as a fully automated deminer. The ability to operated on its own is highly valuable, in the sense that there is no need for a crew maneuvering the vessel on-board, present. Obviously, in the past, a vessel had to be controlled manually. Implementing autonomy, people can be removed from the dangerous area among mines, that in best case can save lives. Hence the possibility of a mine inflicting damage to anyone is absent. Safety is a high priority, and with using automation to help with exactly this, it triggers interest. Figure 1.1 shows the USV Odin located at FFI, while table 1.1 contains some key information about it.

The actuators on Odin is the key element in which it is able to navigate. Controlled in the right manner they can extort force in any direction. The HamiltonJet Water Jets fall in the category of azimuth actuators. An azimuth actuator has the ability to rotate a full 360-degree angle. The actuators mounted by FFI on the USV has the ability to rotate with a 27 degrees angle in both ways from its initial orientation parallel to the

Length Between Perpendiculars (L_{pp})	10.5 m
Mass (m)	6000 kg
Draft Height T	0.7 m
Beam B	3.5 m
Center of Gravity (x_g, y_g, z_g)	(-1.5,0,-0.08)
Center of Buoyancy (x_b, y_b, z_b)	N/A
Actuators	2 x HamiltonJet Water Jet
Rotation of Actuators	(-27°, 27°)
Position of Actuators (x_t, y_t, z_t)	(-4.5,-0.57,0.3), (-4.5,0.57,0.3)

Table 1.1: Available Parameters of Odin

Figure 1.1: FFIs USV Odin

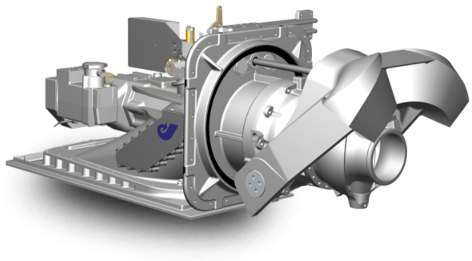

Figure 1.2: HamiltonJet Water Jet. Courtesy of Hamilton Jet

surge axis. Above the actuators there are buckets, that can be lowered from totally above ground to under water. The buckets will reflect the flow of water generated by the propellers, such that the flow will change direction. Combining the two actuators correctly it is possible to maneuver the USV with velocity only in the sway direction (side-ways motions).

Figure 1.2 show a picture of the type of actuator mounted on the USV. The circle at the right-hand side of the figure where the flow is passing through while the USV is in action. The bucket is located above. There are two of the HamiltonJet Water Jet at the rear of the USV, relatively positioned with a distance of negative and positive $0.57\ m$ along the sway axis. The center of the vessel is the reference.

1.4 Literature review

Research done prior in the field of autonomous vehicles are located all around the world. Especially at NTNU there has been a lot of work on the subject. Thor I. Fossen has written a book entirely based on motion control of marine crafts, and various master thesis' has been written on autonomous vessels. The *Department of Engineering Cybernetics* and Norway as a country is a large partaker of the work done on the subject from a global perspective. The first electric and autonomous ship is to set sail in 2018 (Yara Birkeland), is developed in Norway. [1]. Bertram Volker presents a survey of USV with a historical perspective that tracks back to 1944. [2] Most of the USV are from USA. In the 1950s the interest for the use of an autonomous vessel as a deminer or to any other dangerous mission grew, for "obvious reason" he wrote in the report.

Previous master thesis' from NTNU with the likes of *Autonomous Docking for Marine Vessels Using a Lidar and Proximity Sensors* [3] and *Guidance System for Autonomous Surface Vehicles* [4] have been of good help getting a grasp of various problems within autonomy at sea.

A variety of textbook have been useful in understanding and presenting the theory needed.

1.5 Assumptions

It is assumed that the environment around the USV has been mapped by the sensors on the vessel. A map of the surrounding area is generated for the USV to yield a planned path to the pier. The map should point out any objects that could impact the route, such that we can be sure that any path generated by an algorithm would avoid a non-suitable position at sea.

We also assume that the pier is in the first place a well-known pier. It creates an easier task to solve when we have knowledge about the pier. The position and orientation of the pier are known, as well as a designated docking position. In the future, it might be interesting to investigate the opportunities of expanding the solution to unknown piers. In those cases, the USV also need to verify whether there is available space for the USV to park. For this problem, the main goal is for the USV to safely maneuver in and out of the pier located in Horten.

A mode called *Station Keeping* has been developed by FFI. Once the USV is set to station keeping it will remain the current geographical location. It is wished that the USV will enter the station keeping mode at once when the vessel has reached the pier to its desired spot. The final stage of the procedure of the docking mode that is to be explained later will be setting the USV to station keeping. Therefore, we assume that the station keeping implemented by FFI works.

1.6 Background

In the cybernetic field of study autonomous development of vehicles stands central. NTNU students at the master's degree program *Cybernetics and Robotics* are exposed to subjects such as control theory, guidance and control of vehicles and algorithms and data structures, to name a selection of relevant topics. Personally, my knowledge in this area is not large from a practical perspective. As a student, there is more theoretical work than practical work. Nevertheless, it will be interesting to apply theoretical work to practical problems. I am excited to work with the USV at FFI in the months following new years when the master thesis work will begin, with creating and implementing solutions to the USV.

1.7 Outline

The outline of this project report is largely affected by the template produced by NTNU's *Department of Engineering Cybernetics* in how the structure of the report is. Chapter 2 Theory explains the theories useful in solving the problem, while Chapter 3 Concept shows how all the theories can be molded together to form a complete solution. Chapter 3 will consist of several figures explaining the concepts in its entirety. In Chapter 4 Discussion it will in detail be discussed which strategy has best seen fit to solve the problem, as Chapter 3 barely digs into this. Chapter 5 Conclusion is the last chapter of the report. It will contain a summary of the aspects regard the solution. The chapter will also consist of a section called future work, where it will be explained how the work planned for next semesters master thesis will look like.

Chapter 2

Theory

In this chapter, the background theory needed for this project will be presented. The theory that is necessary to get a grasp of the possible solutions of the problem, that is autonomous docking. The first sections consist of a table where the notations of ships motions a present. The following sections present the theories based on each of the sub-problems to this task, previously presented in Chapter 1 Introduction, Section 1.3 Project Description. In the order the sub-problems are presented in the project description is the order of which it will appear in this chapter.

2.1 Notation for Marine Vessels

Table 2.1 shows how we note motions for a marine vessel according to Society of Naval Architects and Marine Engineers (SNAME) [5]. The most important thing to remember is that we denote the x, y and z-direction with surge, sway and heave, and

DOF		Forces and moments	Linear and angular velocities	Positions and Euler angle
1	motions in the x-direction (surge)	X	u	x
2	motions in the y-direction (sway)	Y	v	y
3	motions in the z-direction (heave)	Z	w	z
4	rotation about the x-axis (roll)	K	p	ϕ
5	rotation about the y-axis (pitch)	M	q	θ
6	rotation about the z-axis (yaw)	N	r	ψ

Table 2.1: The notation of SNAME (1950) for marine vessels

the rotation about the surge, sway and heave axis with ϕ, θ and ψ. These notations hold for all maritime vessels.

2.2 Path-planning algorithms

In this chapter, there will be presented a couple of path planning algorithms that could solve the problem. The presented methods will all be able to create a path for a vessel.

All of the various methods are considered as algorithms created to follow a path.

2.2.1 Path Following for Straight-Line Paths

A frequently used method for path following is Line-of-Sight (LOS) guidance. A LOS vector from the craft to the next waypoint or a point on the path between two waypoints can be used for both course and heading control. [6]

We usually consider 2-D horizontal plane motions we applying the LOS steering law.

The LOS steering law is formulated

$$\chi_d(e) = \chi_p + \chi_r(e) \tag{2.1}$$

$$\chi_d(e) = \alpha_k + arctan\left(\frac{-e}{\Delta}\right) \tag{2.2}$$

where χ_d is the desired heading angle, χ_p is the path-tangential angle, while χ_r is a velocity-path relative angle. χ_r ensures that the velocity is directed toward a point on the path that is located a lookahead distance $\Delta(t) > 0$ ahead, and the direct projection of $\imath^n(t)$ on to the path. The lookahead distance is the intersection of e on the path and (x_{los}, y_{los}).

Figure 2.1 shows how the path following problem considering a straight-line path is formed with a LOS steering law. The desired heading is always set to point to (x_{los}, y_{los}).

2.2.2 A^* search algorithm

The A* (read as A star) search is an algorithm that finds the minimum cost path from one point to another by examining nodes in between [7]. The algorithm was first introduced back in 1968 by a group of scientist with the Artificial Intelligence Group of the Applied Physics Laboratory at the Stanford Research Institute in California. The A* algorithm is an extension of the well known Dijkstra's algorithm, that also finds the shortest paths between nodes in a graph. What separates the two algorithms is that the A* algorithm uses heuristics to determine the shortest path. By further explaining a heuristic algorithm, it is an algorithm that ranks the possible alternatives available

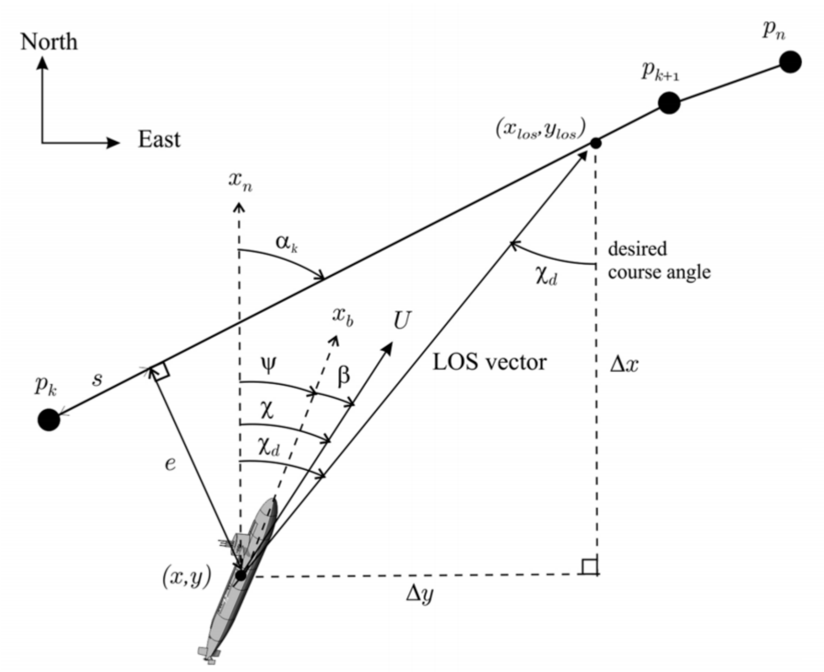

Figure 2.1: LOS guidance where the desired course angle χ_d is chosen to point toward the LOS intersection point (x_{los}, y_{los}). Courtesy of Fossen (2011).

by estimating the cost from the current node we have visited to the goal node.

The A* search selects the path that minimizes the following equation

$$f(n) = g(n) + h(n) \tag{2.3}$$

where $g(n)$ is the distance from the start node to n, while $h(n)$ is the estimated distance from node n to the goal node.

A* is one of the most sought-after algorithms with regards to minimum cost path-finding. Even though, the algorithms have its flaws. One limitation is that it does not always find the shortest path since it is approximating the distance from the current node to the goal node.

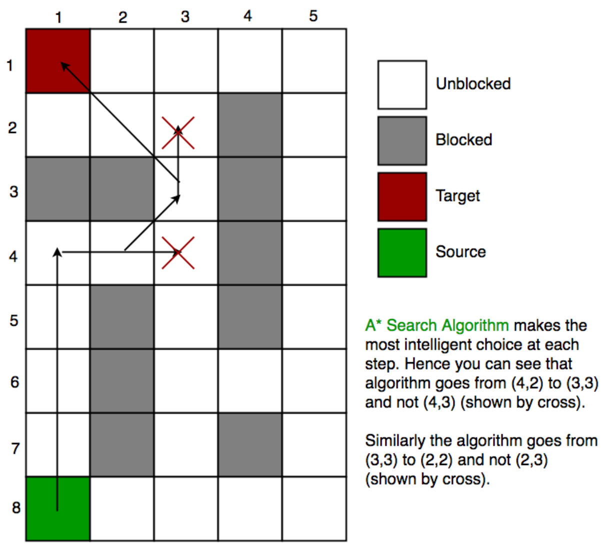

Figure 2.2: A* Search Algorithm example. Courtesy of GeeksforGeeks

Figure 2.2 shows how the algorithm thinks. It makes the most intelligent choice at each step. Looking at the figure we identify the algorithms thinks by it directly going from (4,2) to (3,3) without visiting (4,3). [8]

2.2.3 Rapidly-Exploring Random Trees

Rapidly-Exploring Random Tree (RRT) constructs a tree using random sampling in the search space. Algorithm 1 shows each step. The tree gradually expands as the iteration continues. At each iteration, a random state is selected within the configuration space. If the path between z_{rand} and $z_{nearest}$ is obstacle free, the path between z_{rand} and $z_{nearest}$ is added to the tree. Otherwise, it returns a new node z_{new} by using a steering function, thus expanding the tree from z_{new} to $z_{nearest}$. The algorithm ends when the fixed amount of interactions are executed or the predefined time period expires. The best path is the created from the tree generated. [9]

Algorithm 1 RRT Algorithm
$\mathbf{T = (V, E) \leftarrow RRT(z_{init})}$

```
T ← InitializeTree();
T ← InsertNode(Ø,z_init,T);
for i = 0 to i = N do
z_rand ← Sample(i);
z_nearest ← Nearest(T, z_rand);
(z_new, U_new) ← Steer(z_nearest, z_rand);
if ObstacleFree(z_new) then
T ← InsertNode(z_min, z_new,T);
return T
```

RRT works with nonholonomic constraints, which means that the state of i.e. a USV depends on the path taken in order to reach the goal. It was developed in 1998 by Steven M. LaValle while he was an assistant professor in the Department of Computer Science at Iowa State University. The algorithm will generate a tree structure from the starting node that grows large to unsearched areas in a problem. It does not only calculates the coordinates in the workspace, but also states of the object that a planned path is wished for. [10]

2.3 Feedback Control

When dealing with a dynamical system there are different approaches available to control the system. The most common way is to use any variation of the PID controller. Figure 2.3 displays a feedback control loop where we wish to control the process, P(s). The ultimate goal is to use the controller to eliminate any deviation in the error, $e(t) = r(t) - y(t)$, such that the setpoint equals the process variable y(t). The feedback makes the process variable close to the setpoint in spite of disturbances d and variation of the process characteristics. [11][12]

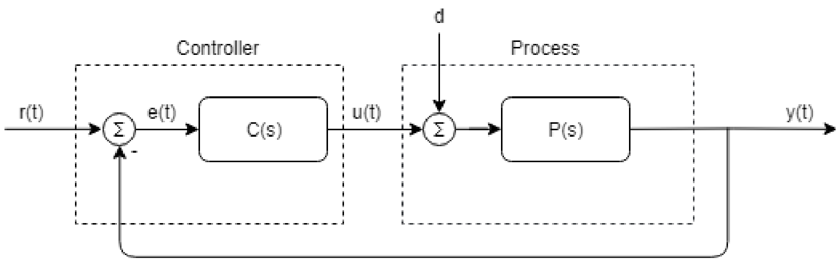

Figure 2.3: Feedback Control Loop

2.3.1 Controller

The mechanism which constitutes to the controller is the PID controller. The PID controller is used in more than 95% of the control loops in process control. It consists of three separate terms: the proportional-, the integral- and the derivative term. Each term affects the behavior of the response we want to control, in various ways. The three terms can be composed in different ways, such that we can have a P-, PI or a PD controller also. In the coming sections, the different terms will be explained, as well as the various constitutions of the various terms.

Proportional term

The proportional term makes sure that any input to a system changes proportionally with the error of the system when looking at the difference between the reference setpoint and output of the system.

$$u(t) = \underbrace{K \cdot e(t)}_{Proportional\ term} \tag{2.4}$$

Integral term

The main function of the integral term (or integral action that it is most known as) will make sure that the steady-state error $e(t)$ agrees with the setpoint $r(t)$. A controller with integral action will always give zero steady-state error, $e(t) = 0$, is tuned correctly. From equation 2.5 the integral term is shown, in addition with the proportional term. The integral action will always be in addition to at least the proportional term.

$$u(t) = K\Big(e(t) + \underbrace{\frac{1}{T_i} \cdot \int_0^t e(t)dt}_{Integral\ term}\Big) \tag{2.5}$$

With the right tuning of the integral action parameter T_i (integral gain), the controller will give the process zero error with respect to the reference setpoint $r(t)$. T_i is a measure of how long time the proportional step in the input takes. The integration goes on as soon as the integral term will be connected to the controller until it eventually gets connected out again. This is often equivalent to the controller being turned on and off.

Derivative term

The purpose of the derivative term (also known as the derivative action) is to improve the closed-loop stability. The derivative terms objective is to create an additional signal to the proportional term such that it can be possible to counteract the changes in the measured process. The derivative term allows for estimation of the future control error. From equation 2.6 the derivative term is shown, in addition with the proportional term. As long as the derivative of the error $e(t)$ is not equal to zero, the derivative term will provide an additional signal. At time T_d ahead, the estimated control error will be equal to the derivative term (equation 2.7). The derivative action will always be in addition to at least the proportional term.

$$u(t) = K\Big(e(t) + \underbrace{T_d \cdot \frac{de(t)}{dt}}_{Derivative\ term}\Big) \tag{2.6}$$

By performing a Taylor series expansion of $e(t + T_d)$ gives

$$e(t + T_d) \approx e(t) + T_d \cdot \frac{de(t)}{dt} \tag{2.7}$$

From the Taylor series expansion, it can be seen that the control signal is proportional with the estimated control error at time T_d ahead. The estimates are obtained by linear extrapolation (estimating signals ahead based on the relationship between the variables).

Equation 2.6 is a PD controller. Shortly explained, the action made by a PD controller may be interpreted as if the control is made proportional to the predicted process output.

2.3.2 Variations of the PID controller

Firstly, we look at the controller with all the term included, the PID controller. From the previous subsections that explain each term we have learned what the effect of each one of them are.

As explained in [11] and [13] , the generalized equation for a PID controller will be

$$u(t) = \underbrace{K}_{Proportional\ term} \cdot \left(e(t) + \underbrace{\frac{1}{T_i} \cdot \int_0^t e(t)dt}_{Integral\ term} + \underbrace{T_d \cdot \frac{de(t)}{dt}}_{Derivative\ term} \right) \tag{2.8}$$

The block diagram of equation 2.8 is represented in figure 2.4. This is how the controller in figure 2.3 will look like if the controller is a PID.

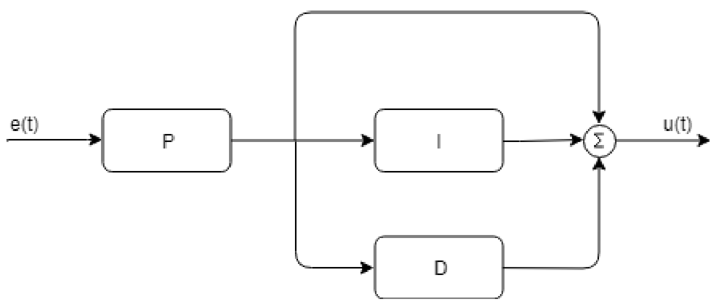

Figure 2.4: Block Diagram of a PID Controller (eq. 2.8)

From equation (2.8) we can see what each part of the PID controller in figure 2.4 contains.

If we manipulate 2.8 we introduce the terms K_p, K_i and K_d, which are the proportional-, integral- and derivative gain, respectively. Equation 2.9 is the end result of the manipulation. When using a common method to adjust the parameters of any PID controller variation (P, PI, PD or PID) called the Ziegler-Nichols method, we calculate the parameter of the three gains K_p, K_i and K_d. This method will be introduced later in this section.

$$u(t) = K \cdot e(t) + \frac{K}{T_i} \cdot \int_0^t e(t)dt + K \cdot T_d \cdot \frac{de(t)}{dt} \tag{2.9a}$$

$$u(t) = \underbrace{K_p \cdot e(t)}_{Proportional\ term} + \underbrace{K_i \cdot \int_0^t e(t)dt}_{Integral\ term} + \underbrace{K_d \cdot \frac{de(t)}{dt}}_{Derivative\ term} \tag{2.9b}$$

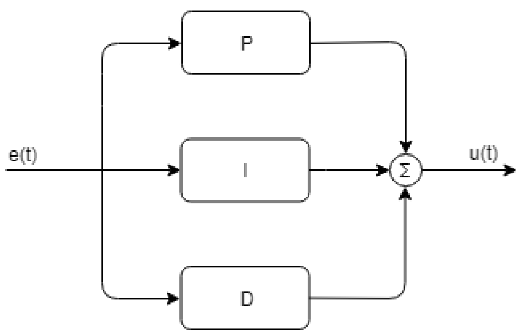

Figure 2.5: Block Diagram of a PID Controller (eq. 2.9b)

P controller

A P controller only includes a proportional gain, K_p. This gain makes sure that any input to the system changes proportionally with the error.

PI controller

If we have an integral term in addition to a proportional term, we have a PI controller. A P controller works very poorly for most systems. The integral terms are added to drive the steady-state error to zero. If the error greater than zero, the integral action will be active, and when the error is equal the zero it is not used (as the integral of zero equals zero). A large value of the integral gain gives a large additional signal from the integral action, while a small value gives a small additional signal. A combination of the proportional- and integral term, the controller can act relatively fast, and push the steady-state error towards zero with right values of the gains. Without a proportional term, the response would have been very slow at the beginning.

PD controller

A derivative term in addition to the proportional constitutes to a PD controller. The derivative action will as explained previously counteract changes in the measured process. At stationary state, the derivative term is connected out, and the controller will works solely as a P controller with exactly the same steady-state error as for a pure P controller. As long as the process is no longer constant, the derivative of the error will be other than zero. When it becomes negative, it is indicated that future error will be positive. To counteract the change in future error, the derivative term will become a positive signal. If the derivative gain is chosen too large it will overcompensate which will destabilize the system. It also dampens the oscillations caused by a large proportional gain.

PID controller

Combining all of the terms we have the PID controller as given by equation 2.9b as the form om the block diagram is shown in figure 2.5. The PID controller exploits both advantages from the integral and derivate term. The integral action removes steady-state error while the derivative action reduces the dynamical deviation and helps counteract changes and oscillations in the measures process. It can be tricky finding the right balance between the parameter gains for a PID controller. As a starting point, the rule of thumb Ziegler-Nichols method can be used to achieve initial values. From there the gains may be tuned for optimal performance.

2.3.3 Ziegler-Nichols method

The Ziegler-Nichols method is a rule of thumb method. With respect to what type of controller we have, there is a fixed table of how the gains should be picked. Table 2.2 shows how it looks like. There are two distinctive parameters that we have to calculate, that tells us something about the dynamics of the control loop. The idea is simple. By setting the integral gain to infinity (a very high number) and the derivative gain to zero, we increase the proportional gain carefully until we have standing oscillations in the process. It is at this moment we attain the parameters in the Ziegler-Nichols table, K_c and T_c (which is the critical gain and critical time period, respectively).

With respect to what controller-type we would like to use, table 2.2 produces the gains of the respective controllers.

	K_p	T_i	T_d
P controller	$0.50K_c$	-	-
PI controller	$0.45K_c$	$0.85T_c$	-
PD controller	$0.65K_c$	-	$0.12T_c$
PID controller	$0.60K_c$	$0.5T_c$	$0.12T_c$

Table 2.2: Ziegler-Nichols tuning parameters

The possibility of needing to adjust the three gains are present. With the control parameters set, we can look at the response of the system to identify with of the three gains that need to be adjusted. An increase of the proportional gain will generally give a faster and more unstable process, and with a decrease, it will give a slower and more stable process. A decrease of the integral action will remove the steady-state error faster, but it will make the process more unstable, while an increase will do the opposite. An increase in the derivative gain will up to a certain level make the process faster and more stable. With a too large T_d it might cause oscillations.

2.3.4 Integral Windup

A controller with integral action combined with an actuator that becomes saturated can give some undesirable effect. In the cases where the control error will be great, the feedback path can be broken. The integrator in the controller may integrate the system up to a very large value (it "winds up"), as the integrator is an unstable system. When the actuators are saturated, it will remain saturated even though the process output might change. If the process output produces measurements that lead to reduces error, the integrator will still be large such that it takes time for the integrator to settle a normal value again. We call this effect integrator windup [11]

Integrator windup can be treated in the following way. By adding limitations to the setpoint variations the controller output may never reach the actuator bonds. This is a conservative approach, that is certain cases will limit the controller performance. This is pure with respect to the actuator, with disturbances not accounted for. With back-calculation and tracking the integral is recomputed when the output saturates. A new value is given at the saturation limit. The controller gets an extra feedback path with back-calculation. Measuring the actual actuator output and forming a new error signal (e_s) as the difference between the output of the controller and the actuator output. This signal is fed to the input of the integrator through a gain (T_t), that is the dynamically reset time of the integrator. This signal is zero when there is no saturation in the actuator output, and thus it will not have any effect on the normal operation. When the new error signal is different from zero (the actuator saturates), the integrator value is driven to zero due to the fact that we connect the new error signal directly to the integrator.

Equations for the control error of the new feedback loop

$$\frac{1}{T_i}e_s + \frac{K}{T_i}e \quad \Rightarrow \quad e_s = -\frac{KT_t}{T_i}e$$

Since $e_s(t) = u_a(t) - u(t)$, it follows that

$$u_a(t) = u_{lim} + \frac{KT_t}{T_i}e$$

where u_{lim} is the saturating value of actuator. Since both u_{lim} and e(t) have the same sign, it follows that $u_a(t)$ always is larger than u_{lim}. Then it is proven that the integrator is prevented from winding up.

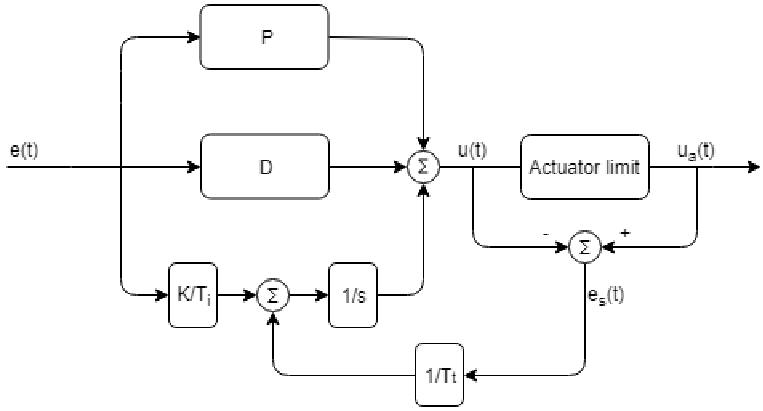

Figure 2.6: Controller with Integrator Windup

2.4 Adaptive Control

Adaptive Control is a method in control theory where the controller which must adapt to a controlled system with parameters that may change. A control law that adapts itself is used in the example of an aircraft that loses fuel under flight, where its mass will change as the fuel in the aircraft decreases. With a lower mass, the system that is the aircraft is better controlled with another set of parameters than when the fuel level of the aircraft was full. A continuous redesign of the controller is in order to avoid inaccuracy or instability of the control system. [14]

If we draw parallels to the problem that is autonomous docking of a USV, the vessel will have different poses from the starting node to the goal node (the pier). The pose of the vessel, that is the combination of position and orientation, will change during its docking procedure. By having multiple parameters that enhance the controller's ability to operate the system, we can achieve a greater result. The speed of the vessel is also of interest. At various speeds, we can set different parameters for this as well.

Figure 2.7 show how the adjustment mechanism would fit into the previous standard feedback control loop. The adjustment mechanism will update the controller with new parameters depending on the measurements that come out of the process. Inside the adjustment mechanism, the control algorithm created for a problem will decide which parameter to use, in our case depending on for example the pose and speed of the vessel.

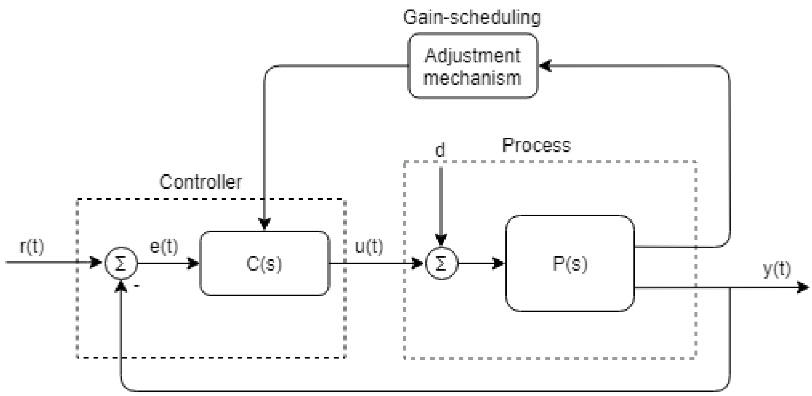

Figure 2.7: Adaptive Controller

2.4.1 Gain-Scheduling

Gain-scheduling is an approach where we have different tuning parameters for different operation points in the system. With the idea that one set of parameters works betters for one case, and set of parameter for another case, we can calculate a great response from the system in every situation. For instance, if we only had one set of parameter for the actuator control we would, in fact, have used the same control system for the entire operation. This may not be the optimal strategy, as the vessel will have different poses and velocities at times. In these situations, one might consider tuning the actuator control system such that a good response is achieved for all the situations the USV may find itself in. When tuning with respect to each situation, we kind of can sew the sets together, forming a variety of sets such that the vessel can handle every situation better. An algorithm containing different criteria based on the pose and velocity of the vessel will the best set to use in the respective situations. The chosen set will be the parameters to the PID Controller that creates the input to the actuators. We remind ourselves that the actuators create the force that can create lateral motions (surge, sway, and yaw). From figure 2.7 we can locate the adjustment mechanism in the control loop. It is in this block the gain scheduling takes place. Figures 2.8 present a simple diagram of how this can be looked at. The chosen set will affect the controller, which is where the input to the process is generated.

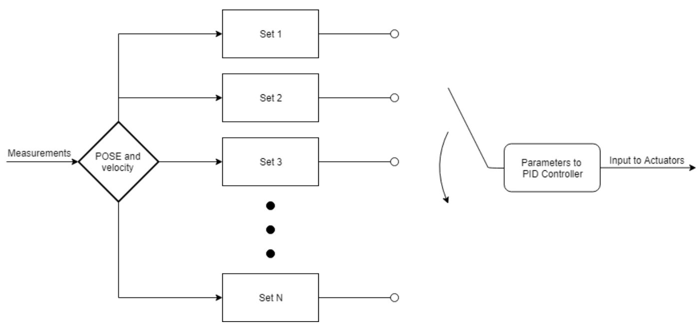

Figure 2.8: The Feedback Mechanism in the Adaptive Control Loop

2.5 Optimal Actuator Allocation

The actuators of the USV are of type Hamilton-Jet Water Jet. There are two of them, both located at the rear of the vessel. They are able to rotate with an angle of 27° both direction from the original position, that is directly in the negative surge direction. Depending on how the actuators are orientated it is possible to create a force in the desired direction. Thrust vectoring as it is called gives us the ability to manipulate the

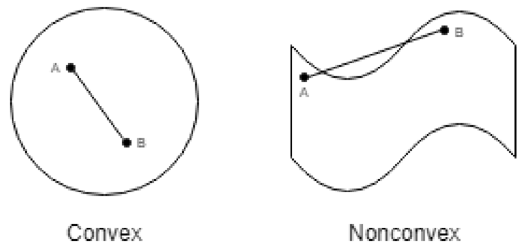

Figure 2.9: Convex and Nonconvex Set

direction of the actuators to create a sum of vectors needed, to control the vessel in the desired manner. The Water Jets are also equipped with a bucket that uses the jet from the water to reflect it, from zero reflection to be able to move the vessel obliquely forward. Combining both the actuator angle and the use of the buckets it is possible to extort force in any direction. Even though the actuators cannot rotate a full 360° angle around its axis, we can call the actuators azimuth since we are able to extort force in any direction using the buckets.

Control allocation of azimuth actuators is an optimization problem that is hard to solve [6]. In the book of Fossen, there is an explanation of *Constrained Control Allocation for Azimuth Thrusters*. For a marine craft equipped with azimuth thrusters/actuators, it generally creates a nonconvex optimization problem. A nonconvex optimization problem is a problem that is harder to solve than a convex problem. From figure 2.9 a convex and a nonconvex set is shown. A nonconvex problem may have multiple local optimal solutions, as we cannot see the whole function. Figure 2.9 explains simply what is meant by that. Point A will not see point B in the nonconvex set as the line drawn between them lies outside the set. An actuator that is azimuth is, in general, a nonconvex optimization problem.

The primary constraint is

$$\tau = \mathbf{T}(\alpha)\mathbf{f} \tag{2.10}$$

where $\alpha \in \mathbb{R}^p$ denotes the azimuth angles. We have to compute the azimuth angles for each sample together with the control input $u \in \mathbb{R}^r$ which are subject to both amplitude and rate saturation. In addition the azimuth thrusters can only operate in a feasible sector that is $\alpha_{i,min} \leq \alpha_i \leq \alpha_{i,max}$ at a limiting turning rate $\dot{\alpha}_i$. Another problem is that the inverse

$$T_w^\dagger(\alpha) = W^{-1}T^T(\alpha)[T(\alpha)W^{-1}T^T(\alpha)]^{-1} \tag{2.11}$$

can be singular for certain values of α. This can cause the maneuverability and dynamic performance of the vessel to be greatly reduced, as the azimuth actuator only can be changed slowly. [15] suggest that the following criterion should be minimized:

$$J = \min_{f,\alpha,s} \left\{ \sum_{i=1}^{r} \bar{P}_i |f_i|^{3/2} + \boldsymbol{s}^T \boldsymbol{Q}\boldsymbol{s} + (\boldsymbol{\alpha} - \boldsymbol{\alpha}_0)^T \Omega (\boldsymbol{\alpha} - \boldsymbol{\alpha}_0) + \frac{Q}{\epsilon + det(\boldsymbol{T}(\boldsymbol{\alpha})\boldsymbol{W}^{-1}\boldsymbol{T}^T(\boldsymbol{\alpha}))} \right\}$$

$$\begin{aligned} \text{subject to} \quad & \boldsymbol{T}(\boldsymbol{\alpha})\boldsymbol{f} = \boldsymbol{\tau} + \boldsymbol{s}, \\ & \boldsymbol{f}_{min} \leq \boldsymbol{f} \leq \boldsymbol{f}_{max} \\ & \boldsymbol{\alpha}_{min} \leq \boldsymbol{\alpha} \leq \boldsymbol{\alpha}_{max} \\ & \Delta\boldsymbol{\alpha}_{min} \leq \boldsymbol{\alpha} - \boldsymbol{\alpha}_0 \leq \delta\boldsymbol{\alpha}_{max} \end{aligned}$$

where

- $\sum_{i=1}^{r} \bar{P}_i |f_i|^{3/2}$ represents power consumption where $\bar{P}_i > 0$ $(i = 1, \ldots, r)$ are positive weights.

- $\boldsymbol{s}^T \boldsymbol{Q} \boldsymbol{s}$ penalizes the error **s** between the commanded and achieved generalized force. This is necessary in order to guarantee that the optimization problem has a feasible solution for any $\boldsymbol{\tau}$ and $\boldsymbol{\alpha}_0$. The weight $\boldsymbol{Q} > 0$ is chose to be large enough so that the optimal solution is $\boldsymbol{s} \approx \mathbf{0}$ whenever possible.
- $\boldsymbol{f}_{min} \leq \boldsymbol{f} \leq \boldsymbol{f}_{max}$ is used to limit the use of force (saturation handling).
- $\boldsymbol{\alpha}_{min} \leq \boldsymbol{\alpha} \leq \boldsymbol{\alpha}_{max}$ denotes the feasible sectors of the azimuth angles.
- $\Delta\boldsymbol{\alpha}_{min} \leq \boldsymbol{\alpha} - \boldsymbol{\alpha_0} \leq \Delta\boldsymbol{\alpha}_{max}$ ensures that the azimuth angles do not move too much within one sample, taking $\boldsymbol{\alpha}_0$ equal to the angles at the previous sample. This is equivalent to limiting $\dot{\boldsymbol{\alpha}}$, that is the turning rate of the thrusters.
- The term

$$\frac{Q}{\epsilon + det(\boldsymbol{T}(\boldsymbol{\alpha})\boldsymbol{W}^{-1}T^T(\boldsymbol{\alpha}))}$$

is introduced to avoid singular configurations given by $det(\boldsymbol{T}(\boldsymbol{\alpha})\boldsymbol{W}^{-1}\boldsymbol{T}^T(\boldsymbol{\alpha})) = 0$. To avoid division by zero, $\epsilon > 0$ is chosen as a small number, while $Q > 0$ is a scalar weight. A large Q ensures high maneuverability at the cost of higher power consumption and vice versa.

The optimization problem suggested by [15] is a nonconvex nonlinear program, which indicated that the amount of computations effort is rather large at each sample, as explained in [16].

2.6 Spike Detection

On the USV there is an accelerometer. The accelerometer measures the acceleration of the vessel. From the measurements, the signal can be handled such that it is possible to detect whether the USV has reached its final position. We wish to receive feedback when the vessel has reached the pier. The very last phase of the vessel consists of slow horizontal motions (in sway direction). We assume that the speed of the vessel at this point is close to constant, as it continuously drifts. This leads to a close to zero acceleration of the vessel. Using a Simple Moving Average (SMA) algorithm upon the signal we can analyze the data to verify when the vessel has reached the goal. In theory, the signal should show a spike in the data at the moment the USV makes contact with the pier.

By creating an algorithm that detects when a spike in the accelerometers sway direction occurs, we can get feedback at the time of occurrence and tell the vessel to stop. Using a moving mean average on the acceleration in sway direction it can be detected when the spike occurs.

Moving Average Filter
The SMA algorithm computes the average (mean) value over a specified number of periods. Whilst the time-series of data is created, the moving average gets updates as fast as a new value is present. Since the motion of the vessel may not be constant, the moving average is a nice way of computing the average value of the acceleration such that an abnormality in the signal easily can be detected. At the moment of time we get notified that a spike has occurred, we immediately know that the USV has reached the pier.

Moving average is the most common digital signal processing filter because it is the easiest filter to understand and use. [17] The moving average filter operates by averaging a number of points from the input signal to produce each point of the output signal. The equation of the moving average can be written as

$$\bar{y}_{MA}[i] = \frac{1}{M} \sum_{j=0}^{M-1} x[i-j] \tag{2.12}$$

where an arbitrary output could look like

$$\bar{y}_{MA}[10] = \frac{x[10] + x[9] + x[8] + x[7] + x[6]}{5} \tag{2.13}$$

When calculating values live, the moving average gets updated with the new value while the last value is removed. The value of n is how many samples the moving average take into account when it is calculated.

$$\bar{y}_{MA} = \bar{y}_{MA,prev} + \frac{y_{MA}[i+1]}{M} - \frac{y_{MA}[i-1]}{M} \tag{2.14}$$

This form of moving average is the simple moving average. There also exists an exponential moving average filter, that weight the most recent data points more.

Chapter 3

Concept

This chapter consists of the concept that is thought to solve the task in hand. The chapter will consist of elements previously discussed in Chapter 2 Theory, without specifically calling out which methods that are a part of the possible solution. Even though, it will be possible to recognize where the theories will fit in to from the approach. It will be possible to imagine for yourself which theories that can solve different part of the total process. In the following Chapter 4 Discussion there will be presented why specific theories will be used to fulfill the concept. The different aspects of the concept that can be connected in order to achieve docking will be visualized. The concept will be tried out, tested and implemented as of the following semester. The achieved results will then be presented in the master thesis report. The concept is only a possible solution.

As mentioned in the section of the introduction the case is to construct an algorithm that allows the USV to autonomously dock at a pier. In the first place, we create a concept that visualizes how autonomous docking can be achieved. The same concept can be transformed into a problem where we wish to take off from the pier. It is a much easier task to solve, and therefore we focus mostly on the autonomous docking part of the problem. If the USV succeeds to dock, it will very likely also succeed in leaving the pier with some adjustments to the code of the docking problem.

The first phase of the problem consists of docking the USV at a known entity, with the option to further develop the docking to an unknown pier. The location of the known pier is in the surroundings of FFIs office in Horten. If this is easily done, we can then consider expanding the algorithm to be able to dock at any pier. The difficulty of docking a pier at an unknown location increases as there is a lack of knowledge of the surrounding area. In these cases, we rely heavily on the sensors on the vessel to map the area. It is especially important to map the pier good enough and also detecting where it is space to dock.

The concept totally consists of several operations, that in the end can be called the process that is docking mode. To start with we need to plan a safe route between the current position of the USV and the pier. Regardless of the orientation of the USV we wish to plan a path, where we can use a variety of path-planning algorithms. From section 1.5 in chapter 1 it is mentioned that a generated map is assumed to be good enough to handle obstacle detection. With this assumption, a path-planning algorithm can be placed upon this map to find the shortest and easiest path to the goal, which is the pier. With the path to the pier planned we also want also need the vessel to dock with a specific orientation, that is the USV parallel to the pier. How the USV is supposed to achieve the desired heading angle so that the USV is aligned with respect to the pier can vary. The concept is presented in three separate figures, where the differences is how the USV reaches the desired heading angle.

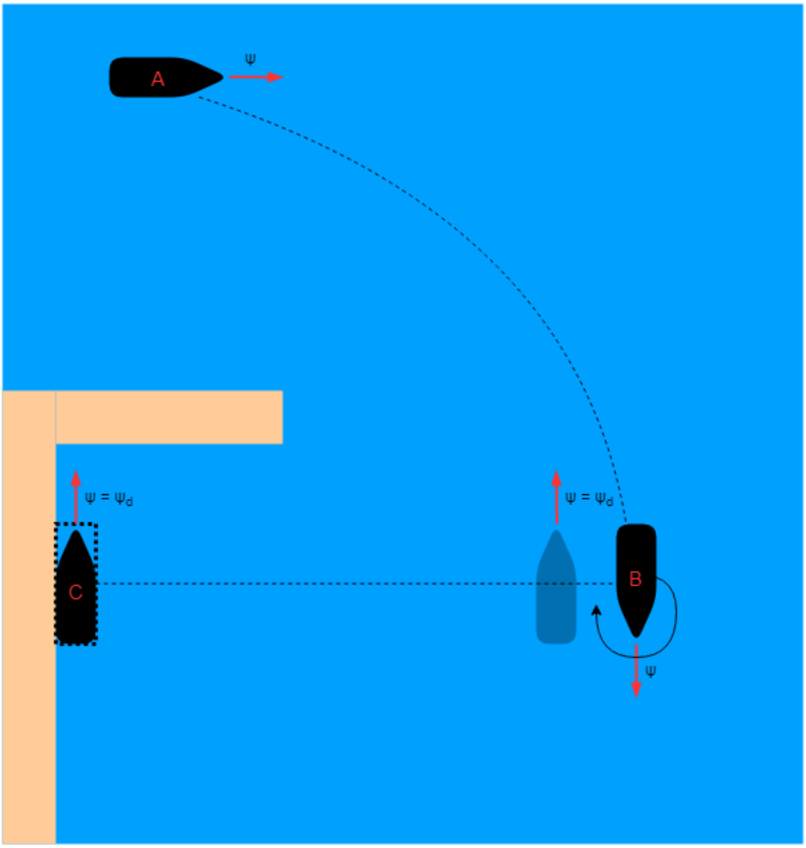

Figure 3.1: Concept 1. Sketch of the event sequence where the USV achieves desired heading by a yaw rotation at a fixed point B parallel to the pier

Figure 3.1 show a situation where we wish to dock the USV. At a random position at a distance from the pier, we have the vessels initial position (point A). From this position, we wish to create a safe path from the initial position and all the way to the desired docking spot. A path-planning algorithm generates a path for the USV, where collision management also is accounted for. The path must travel by a fixed point at a distance from the pier. At this point, it is thought that the actuators can be regulated in order to achieve the desired heading angle. The vessel rotates at point B, with nothing else in mind than achieving the desired heading angle. With the desired heading in place, we wish to slide the vessel into the docking spot by maintaining the desired heading angle. This is possible due to the actuators ability to rotate with an angle both ways and with use of the buckets. We maintain the desired heading by controlling the actuator inputs such that the vessel only has speed in the sway direction.

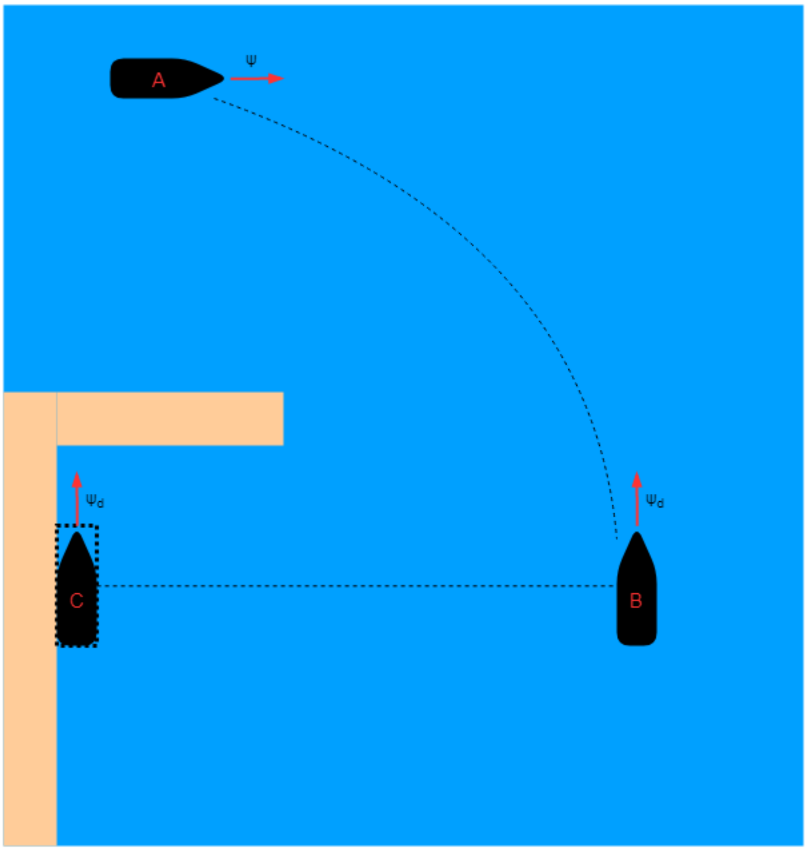

Figure 3.2: Concept 2. Rotation of the yaw angle while moving from point A to B

Concept 2 is very similar to concept 1. In this concept, we look at the possibility of achieving the desired heading angle as we move from point A at a distance from the pier, to point B at a distance parallel to the docking spot. It can be looked like a continuous motion from point A to B, and thus it can get from point A to C faster. It may not be optimal from certain poses of the vessel. Therefore we implement a criterion in the algorithm that says we can start the motion when we are in a specific pose near the fixed point at a specified distance from point C (point B).

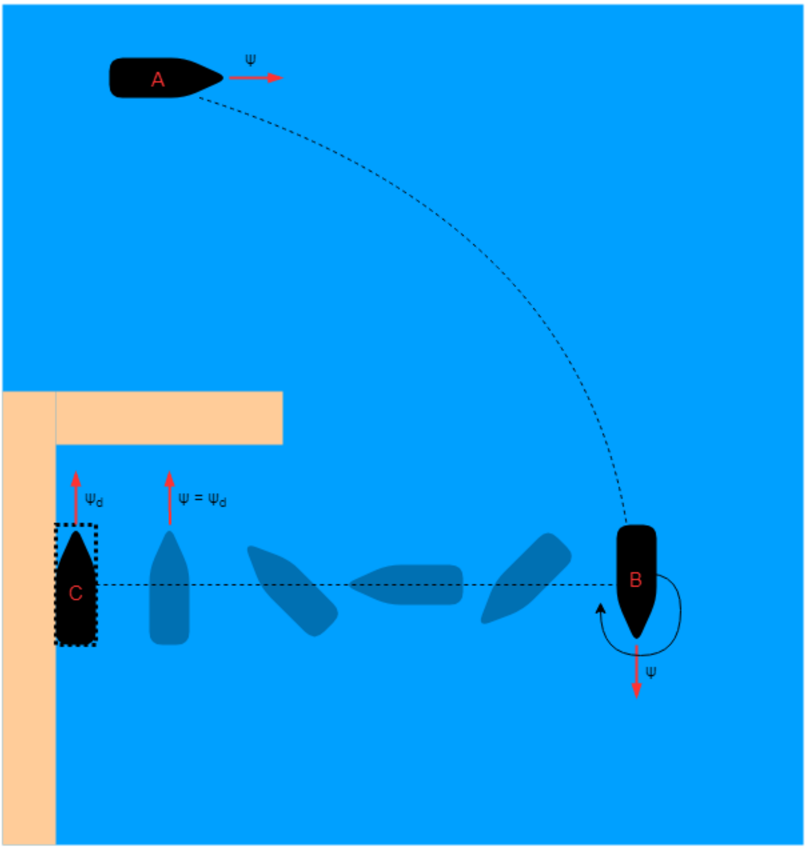

Figure 3.3: Concept 3. Rotation of the yaw angle while moving from point B to C

Concept 3 looks at the possibility of rotating the yaw angle while moving from point B to C. With this concept we can use the force that is created to attain desired heading angle to push us closer to the designated docking position. This looks like the most suitable approach, as we also know that we are in a safe area close to the pier. By using the actuators force we would, in theory, be able to drift to C. If this is the approach it would be necessary to have the vessel rotated as a small distance prior to entering the docking position. This is because we wish to have a stable heading angle and to prevent having the vessel over-rotating.

In the last phase of every concept, we assume the USV to have the same pose. At a close distance to the pier, the last element of the operations consists of sliding into the desired position and detecting when the USV is in place. By using a moving average filter upon the acceleration data of the vessel, it is possible to detect whether the vessel has reached the pier. When contact is made, an abnormality in the acceleration data will happen. The moving average filter creates a signal we could compare with the actual measurements of the acceleration. At the moment in time the USV has made contact with the pier, a spike in the data can be detected by comparing the two signals. If the value reaches a specified threshold that needs to be measured prior, a spike is detected (an abnormality in the signal). As the spike is detected, the USV is set to station keeping, which is the final stage of the docking procedure.

3.1 Flow Chart

The concept is in figure 3.4 represented in the form of a simple flowchart. The flowchart describes which actions that need to happen before another in order to reach the desired position. The generated path to the docking position needs includes point B, where we will make sure that the heading angle is secured one way or another. From point A to B the vessel operates as it normally would out at sea, where the speed and orientation of the vessel force it to point B. From B to C the desired heading angle is maintained or achieved before the USV slides into place.

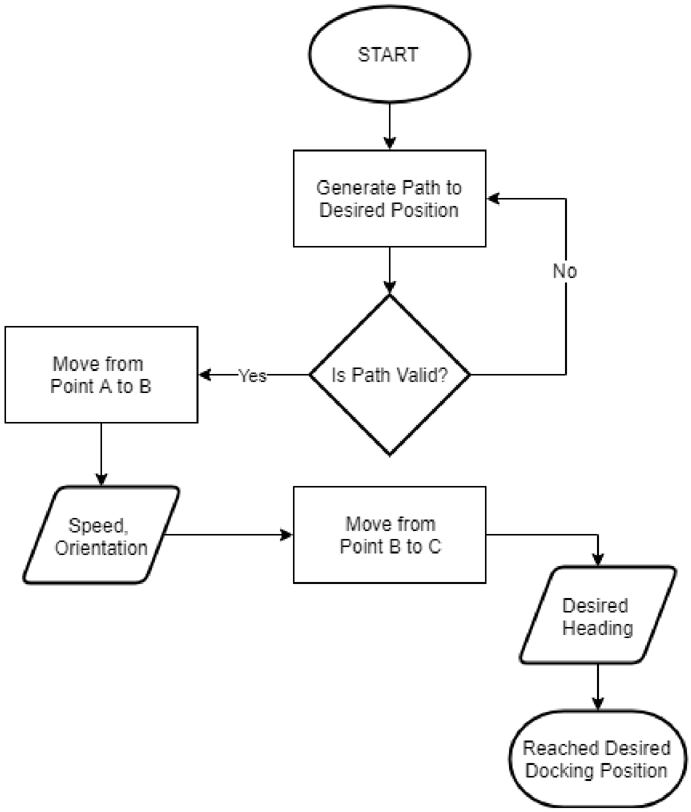

Figure 3.4: Concept in the form of a flowchart

Chapter 4

Discussion

In this chapter, it will be presented which theories that are seen best fit to be implemented in the concept presented in Chapter 2 Concept. As the problem can be broken down to three different tasks, where the theories with the highest probability of success will be presented accordingly.

4.1 Path-Planning Algorithm

To create a path for the USV with regards to the docking and departure procedure, a path planning algorithm will be implemented to achieve this. In Chapter 2 the following path-planning algorithm were presented: LOS Steering Law ,A* and RRT.

The LOS steering law uses equation 2.2 to follow a straight line. Using this method we would have to put out waypoints for it to work. The steering law follows will make the vessel track each line in between the waypoints by controlling the heading angle based on the lookahead distance set. The distance between points is not large, such that it is assumed that all waypoints of the path can be seen by the USV. Therefore the USV will have the next waypoint directly as a target.

The A* search algorithm could be used as the problem is of the *One Source - One Destination* kind. The source is either the pier or a point in the surrounding area. With the map of the surrounding area, it can be broken down into consisting of small squares. The A* algorithm should suit very well to our task in hand. It should not be the most challenging part of this problem finding the optimized path for USV between a node in a distance from the pier, to the goal node at the pier. As understood from the figure 2.2 in section 2.2.2, it will be useful to produce a similar figure by specifying available and occupied nodes (blocked and unblocked in figure 2.2). Finally, a path is drawn between the starting node to the goal node.

The RRT algorithm uses expanding trees to create a path. The algorithm will run for a selected amount of iteration or runtime. The tree grows larger as the iteration increases, and the possibility of finding a better path increases with the number of iterations. Combining the two algorithms may work well. A* creates the path while RRT looks at the vehicle dynamics to create the possible path for the USV to follow with respect to how the USV can maneuver. For instance, we can not enter the goal node in a perpendicular orientation with respect to the pier, as this is not how we wish the USV to park. We have to look at how the orientation of the vessel should be, and with that creating a possible route for the USV to drive. The constraints for certain parts of the algorithm should be the heading angle for instance.

4.2 Feedback Controller

The need for a controller to handle the heading angle of the USV is present. The actuators need to be able to reach and maintain the desired heading angle in the docking procedure. For the departure procedure, we start with the desired heading and wish to keep the desired heading angle as long as needed until the vessel is a distance far from the pier in open water where it can operate further. It is the actuator control that translates to the vessel's ability to follow the created path. Firstly, giving us the needed power to reach the positions, and secondly, dispatching the power in the right directions such that the vessels orientations is as specified. It is crucial that the USVs heading is correct, otherwise, we cannot say that the USV has parked.

For instance, we wish to control an unmanned vehicle to maintain a specific heading angle. The action that is needed to control in order to maintain the heading angle would be the yaw-angle of the unmanned vehicle. The input of the actuators will be the parameters that could change the yaw-angle. The PID controller will give the actuators the input needed in order to get the wanted response, that is to reach the desired destination.

This will work with a traditional PID controller. The integral term eliminates any deviation in the error while the derivative term reduces the dynamical deviation and helps counteract changes and oscillations of the process. Tuning the controller with the Ziegler-Nichols method will give us parameters that should work fine, but the need for readjustment tuning will be present.

4.3 Spike detection

The SMA filter will be used to detect when the USV has reached the designated docking position. It will only be necessary to implement this filter in the docking procedure, as we have no need for it when departing. Once a spike in the accelerometer data has appeared, the vessel can be set to station keeping mode. The SMA should be fairly simple to implement once the threshold has been given value after testing.

Chapter 5

Conclusion

In this chapter, there will be concluded how the concept in its whole can be formed with the theories presented, as well as what to go for. In the previous chapter, it was discussed how the theories in chapter 2 could be used in the concept. As nothing has been tried out physically, this may or may not work as planned.

Concept 3 is seen as the best concept, with respect to the ability to use the force of rotation to move the vessel in the right direction.

Again looking at the project description we remind ourselves of the following problems needed to be solved. The first task consists of creating a path from point A to point C. An algorithm of the form of A^* will most likely be feasible. The surroundings of the pier are not demanding in the form of objects located all around the place.

The controller will initially have the form of a PID. Time will tell whether a PI controller also can do the job. Even though, the PID controller is a better version of the PI and should give the same output or better.

In the last phase of the docking procedure, the SMA filter can detect when the USV is in place.

Creating an optimal allocation of the actuator inputs is not necessary is the sense of creating a solution that handles docking and departure. It can be a nice feature added to the working solution in the end. Since it can be a tricky problem to solve on its own, the focus lies mostly on the actual movement from a point A a close distance in the surroundings of the pier, to the docking spot a point C.

5.1 Work planned

The work planned for the master thesis semester will consist of the creation and implementation of the concept presented in this project report. The concept explained in chapter 3 may not be the most optimal solution, but it will be a foundation for the work the following semester. The theories and ideas presented will be used.

In the situation where the parameter set of the regulator changes we might observe instability in the time period until the new set takes the action. It is wished to investigate this further when testing a parameter change on the vessel to see if this will need to be adjusted.

Acronyms

COLREGs International Regulations for Preventing Collisions at Sea. 4, *Glossary:* COLREGs

FFI Norwegian Defence Research Establishment. i, 2–4, 8, 36

LOS Line-of-Sight. 12, 13, 43

NTNU Norwegian University of Science and Technology. i, 2, 6–8

PID Proportional, Integral, Derivative. 17, 23, 28, 47, *Glossary:* PID

RRT Rapidly-Exploring Random Tree. 16, 43, 44

SMA Simple Moving Average. 31, 32, 46, 48

SNAME Society of Naval Architects and Marine Engineers. 11

USV Unmanned Surface Vehicle. ii, vi, 2–4, 6–8, 16, 26, 28, 31, 32, 35–37, 40, 41, 43–46, 48, *Glossary:* USV

Glossary

COLREGs Published by the International Maritime Organization (IMO) and set out, among other things, the "rules of the road" or navigation rules to be followed by ships and other vessels at sea to prevent collisions between two or more vessels. 4

PID A controller used in control theory to control a process that we want to make sure behaves in a desired manner. Read 2.3.1 for further explanation. 17

USV A vessel that has the ability to operate autonomously without the need of any crew on board. ii

Bibliography

[1] Kongsberg Gruppen. Autonomous ship project, key facts about YARA Birkeland. https://bit.ly/2zNnm4T.

[2] Volker Bertram. Unmanned Surface Vehicles – A Survey. 2008.

[3] Joachim Spange. *Autonomous Docking for Marine Vessels Using a Lidar and Proximity Sensors, MSc thesis.* 2016.

[4] Thomas Stenersen. *Guidance System for Autonomous Surface Vehicles, MSc thesis.* 2015.

[5] SNAME. Nomenclature for Treating the Motion of a Submerged Body. *Technical and Research Bulletin*, (1-5), 1950.

[6] Thor I. Fossen. *Handbook of Marine Craft Hydrodynamics and Motion Control.* John Wiley & Sons, 1st edition, 2011.

[7] Peter E. Hart, Nils J. Nilsson, and Bertram Raphael. A Formal Basic for the Heuristic Determination of Minimum Cost Paths. *IEEE Transactions of Systems Science and Cybernetics*, 4(2):100–107, 1968.

[8] Rachit Belwariar. A* Search Algorithm. https://www.geeksforgeeks.org/a-search-algorithm/.

[9] Iram Noreen, Amna Khan, and Zulfiqar Habib. A Comparison of RRT, RRT* and RRT*-Smart Path Planning Algorithms. *IJCSNS International Journal of Computer Science and Network Security*, 16(10), 2016.

[10] Steven M. LaValle. Rapidly-Exploring Random Trees: A New Tool for Path Planning. 1998.

[11] Karl J. Åström and Tore Hägglund. *PID Controllers: Theory, Design and Tuning*. Instrument Society of America, 1995.

[12] M Araki. PID Control. *Control Systems, Robotics, and Automation*, II.

[13] Kåre Bjørvik and Per Hveem. *Reguleringsteknikk*. Kybernetes Forlag, 2014.

[14] K.J. Åström. *Adaptive Control Theory*. 1988.

[15] Tor A. Johansen, Thor I. Fossen, and Svein P. Berge. Constrained Nonlinear Control Allocation With Singularity Avoidance Using Sequential Quadratic Programming. *IEE Transaction on Control Systems Technology*, TCST-12:211–216, 2004.

[16] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer-Verlag, New York, NY, 1999.

[17] Steven W. Smith. *The Scientist and Engineer's Guide to Digital Signal Processing*. CaliforniaTechnical Publishing, second edi edition, 1999.