CHALMERS
UNIVERSITY OF TECHNOLOGY

Enhanced Auto Pilot for Marine Applications with Adaptive Speed Control

Master's thesis in Electrical Engineering

ERIC KARLSSON PONTUS LUNDBERG

Department of Electrical Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2017

Master's thesis 2017:EX070

Enhanced Auto Pilot for Marine Applications with Adaptive Speed Control

ERIC KARLSSON
PONTUS LUNDBERG

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Electrical Engineering
Systems and Controls
Chalmers University of Technology
Gothenburg, Sweden 2017

Enhanced Auto Pilot for Marine Applications
with Adaptive Speed Control
ERIC KARLSSON
PONTUS LUNDBERG

Supervisor: David Nydahl, CPAC Systems
Examiner: Torsten Wik, Electrical Engineering Chalmers University

Master's Thesis 2017:EX070
Department of Electrical Engineering
Systems and Controls
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Enhanced Auto Pilot for Marine Applications with Adaptive Speed Control

ERIC KARLSSON
PONTUS LUNDBERG
Department of Signals and Systems
Chalmers University of Technology

Abstract

Today's society is highly focused on safety and environment, this together with ride comfort are used as selling points for autonomous cars. Boats are also vessels commonly used in traffic and therefore have similar rules and guidelines to consider, as well as being seen as a symbol of luxury and a comfortable life. Autopilots have been used in the marine sector a long time, but only to keep bearing of the boat. This project's aim is to implement a speed controller for boats to be used together with existing autopilots, to reach a higher level of comfort and also reduce the number of stressful tasks for the captain.

A simulation environment is build up in Simulink and Matlab to be able to test and develop the controller without having to do sea trials. In the simulation environment a physics engine is built to represent a boat's movement in six degrees of freedom where also two different boats are modeled. Instead of using existing autopilots in the simulation environment a new steering algorithm has been developed.

In simulation the speed controller showed great potential, especially in more curvy routes. It managed to increase comfort and path following ability without increasing the run time to finish the route. However, while doing sea trials and using existing autopilots to control the heading of the boat the result was very different.

From the results of the sea trials it is concluded that today's autopilots are not advanced enough to run a multi point route with active speed control. Instead, the steering and speed controls would benefit from being developed in conjunction with each other.

Keywords: Autopilot, Enhanced Speed Control, Vessel Simulation

Acknowledgements

We would like to express our graditude to CPAC Systems for allowing us to do the thesis at their company, but also for all the support, data and guidance. A special thanks to our supervisor at CPAC, David Nydahl, who has guided us and provided feedback and support throughout the entire project. We would also like to thank our examinar, Torsten Wik for helping us complete the thesis. Last, we would like to thank Volvo Penta for providing us with data for the project.

Eric Karlsson, Gothenburg, 2017
Pontus Lundberg, Gothenburg, 2017

Contents

List of Figures

List of Tables

Nomenclature

α	Angle between equilateral triangle of the simplified hull.
β	Angle between resulting lateral acceleration and gravity.
$\ddot{\omega}$	Yaw acceleration.
$\ddot{X}, \ddot{Y}, \ddot{Z}$	Second time derivative of X, Y, Z.
$\ddot{x}, \ddot{y}, \ddot{z}$	Second time derivative of x, y, z.
δ	Rudder angle.
$\dot{\omega}$	Yaw velocity.
$\dot{m}$	Mass flow from the propeller.
$\dot{X}, \dot{Y}, \dot{Z}$	First time derivative of X, Y, Z.
$\dot{x}, \dot{y}, \dot{z}$	First time derivative of x, y, z.
η	Efficiency of propeller
γ_1	Constant for calculating rate of change of RPM depending of throttle.
γ_2	Rate of decrease of motor RPM for 0 throttle.
Ω	Average rotation velocity of the propeller.
ω	Yaw angle, the angle around z.
ρ	Density of water.
τ	Torque.
a	Acceleration.
A_{rudder}	Area of the rudder.
$Area$	Reference area for calculating fluid dynamic forces.
B	Instantaneous waterline on the vessel.
B_0	Width of the waterline on the vessel while in rest.
C_D	Coefficient of drag.
C_L	Coefficient of lift.
C_{x2}	Axial velocity of the water out of the propeller.
CD_A	Specific drag constant.
D_{out}	Derivative term in the PID controller.
$e(t)$	Error in the PID controller.

F_{fluid_i} Force exerted from the fluid in the i-direction.

$F_{resulting}$ Resulting yawing force around the rotation axis.

h Instantaneous submerged depth.

h_0 Submerged depth of the vessel while in rest.

$heading$ Heading direction of the vessel.

$I_i i$ Inertia around the ii-axis.

I_{out} Integral term in the PID controller.

K_d Derivative parameter in PID controller.

K_i Intergral parameter in PID controller.

K_p Proportional parameter in PID controller.

L Angular momentum.

l Length of the lever arm between the rotation axis and the resulting force.

m Mass of the vessel.

m_i Mass plus the displaced mass in direction i.

P_{out} Proportional term in the PID controller.

$pitch$ Pitch of the propeller.

$pods$ Number of pods or propellers on the vessel.

R Gearratio of the motor.

r Radius of circle.

$r(t)$ Setpoint input to PID controller.

r_{prop} Radius of the propeller.

$slip$ The slip ratio of the propeller.

t Time.

T_u Oscillation period.

$u(t)$ Modified input signal to the PID controller.

V_{prop} Velocity of of water in the propeller.

X, Y, Z Global coordinate system.

x, y, z Body fixed coordinate system.

$y(t)$ Output from PID controller.

CoB Centre of bouyancy.

DOF Degrees of freedom.

dRPM Rate of change of revolves per minute.

$\mathrm{F}_{throtti}$ Propalsional force from the drivetrain in direction i.

g Gravitational constant.

PID Proportional, Integral and Derivative regulator.

roll	Angle around x axis.
RPM	Revolves per minute.
v	Velocity of the vessel.

1
Introduction

The project presented in this report is a pilot study which aims to further develop marine autopilots. The current autopilots only control the steering system while leaving the throttle in control of the captain. The idea is to expand such autopilots to improve the ability to follow a certain path by also controlling the throttle.

1.1 Background

Today's society is highly focused on safety and environment, the automotive and marine industries are no exceptions. Often when talking about safety, the human error is a large factor, especially when looking over an expanded time range. People are often considered to be safe drivers during short time periods, but after longer periods they may become susceptible to fatigue and lose the ability to concentrate and make rational decisions to a certain extent. The reason is that many there are variables and inputs to be keep track of at the same time. At any given point the driver of a vessel needs to control the vessel, i.e. keep the selected path and speed but also keep track of the surroundings, e.g. islands, shallow waters, buoys, and other boats. By minimizing the number of tasks a driver needs to execute, the focus can be more concentrated on just a few tasks, and thus lowering the risk of driver fatigue. One way to reduce the number of tasks is to implement an autopilot. In this way the driver can focus more on keeping track of the surroundings and less of the vessel [2].

Another possible advantage of using an autopilot is the environmental benefit. Autopilots of today are already better than humans on keeping the vessel on the correct path. This means the autopilot can shorten the cruise time and thereby also reduce the fuel consumption [4] [3].

Existing navigation systems, e.g. plotter systems, can predict and calculate the route. As plotter systems become more advanced, the more detailed routes can become. Current advanced systems can create routes from one point to another, while avoiding obstaces such as islands and shallow water. Together with an advanced autopilot it is possible to create an autonomous experience in open waters on a route with moderate curvature. However, when the route becomes more complicated, the captain still needs to take command of the throttle.

1.2 Purpose

The purpose of this project is to create an addition to autopilots, which adapts the velocity of the vessel for a given route, taking into consideration the path and the comfort of the passengers. Furthermore, the autopilot should be general in the sense that it should work for different types of boats and for different autopilot steering systems.

1.3 Limitations

Prior to the project, the following limitations were agreed upon:

- External factors, such as waves, are not taken into consideration.
- The model is only tested on power boats between 30-100 ft.
- The focus of the project is speed adaptation in corners.
- The simulation environment is kept as simple as possible to put more focus on the actual speed control.

1.4 Problem Formulation

The project involves multiple different subproblems to reach the purpose. In order to effectively develop a model for speed regulation a simulation environment needs to be set up, before eventually testing it in a real life environment.

1.4.1 Simulation Environment

The first problem is to set up a simulation environment where the algorithm can be tested. The extent of the real life testing, however, is limited due to time and monetary constraints. The virtual environment needs to be adequately accurate in order for simulations to be valuable. This could either be achieved by the use of an existing model or by creating a new model. The accuracy of the selected method needs to be validated by real world testing for at least one vessel, in the order of dimension of interest.

1.4.2 Waypoints to Trajectories

The path which the vessel should follow will be given in the form of waypoints. The waypoints are then converted into a trajectory that the vessel should follow. The minimum turning radius needs to be sufficiently large for the vessel to be able to turn. The trajectory also needs to be optimized to be as comfortable as possible while still going adequately fast.

1.4.3 Steering

A controller needs to be implemented in order for the vessel to keep the correct heading as given by the trajectory. The steering algorithm also needs to be able to turn with a minimum and a maximum turning radii. Furthermore, the steering algorithm and the speed estimation algorithm need to work independently.

1.4.4 Speed Estimation

As mentioned, a model for prediction of speed with respect to route tracing and comfort needs to be developed. One of the main issues then is that the model needs to be fully modular, meaning that it should be independent of what vessel it is implemented in. The main issue is most likely not compatibility issues, but rather that since it is a marine vessel, it is not granted that a purely reactionary model will work, due to the fact that for different motors the delay from change in torque from the motor to change in vessel acceleration can vary greatly. Since very little data is available, the key point parameters need to be identified for both tracing a set route and comfort. This involves finding the required signals needed in the vessel, that is the needed sensors for running the model, and also to quantify the comfort.

1.4.5 Litterature Study

Before starting to build up the project work a litterature study was carried out to see what had already been done within autonomous sea vessels and autopilots for boats. Most of the litterature that was found was from previous master thesises at Norwegian University of Science and Technology (NTNU). The main topics for these projects were, the best way to combine waypoints in a drivable route [5] and a comparison between two different observers for dynamic positioning on open water, which of one utilizes a vessel model while the other does not [7]. Both projects used a GPS to, in one case keep the boat stationary and in the other case follow a certain path autonomously. The next step after this, the step that was done in this project, was to implement a speed controller which took both comfort and arrival time in consideration.

1.4.6 Results

The correlation between simulation and testing was sufficiently good, the main difference between the simulation results and the testing results was due the fact that two different steering algoritms were used. In the real world testing a commercial steering algoritm was utilized and in the simulations a self-developed steering algoritm was used. This ment that the vessels behaved differently. The reason for not using the same steering algoritm was due to the fact that the project was ment as an enhancment to auto pilot systems, rather than a complete new one. The main result from this was that without a sufficiently good steering system, it will not be possible to enhance the path following ability to a satisfactory level.

2

Theory

Two different coordinate systems have been used throughout the project. One global, represented by capital letters, X, Y, Z and one body-fixed, represented by lower case letters, x, y, z. The global coordinate system is with respect to the global coordinate axis and can be compared to a GPS system, with the X-axis pointing north, the Y-axis is pointing east and the Z-axis pointing vertically downwards. The body-fixed system is an inertial coordinate system with the origo in the center of the vessel, the x-axis always pointing in the direction of the boat, the y-axis pointing perpendicular to the right of the direction of the vessel when the boat is leveled with the water and the z-axis is vertically downwards from these two, the rotations around x, y and z axis are called roll, pitch and yaw (see Figure 2.1).

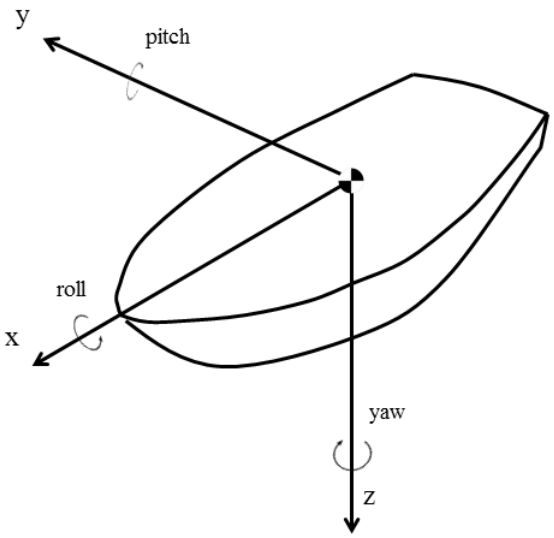

Figure 2.1: Inertial coordinate system.

2.1 Physics Model

The physics model is a 6 degrees of freedom (DOF) model consisting of three translational and three rotational. A quasi-static steady-state is assumed and the equations of motion for each DOF are solved.

Since creating a simulator was not the primary focus of the project, the following assumptions were made:

- The aerodynamic effects are assumed to be negligible.
- There are no external forces, i.e. no currents, winds etc.
- The displaced water mass is constant.
- There areno collision.
- Drag coefficient (C_D) and lift coefficient (C_L) are constant and not dependent on the rotation of the vessel. Furthermore the vessels does not go into plaining.
- All rudders/pods have the same steering angle.

The solver used in the physics model is an iterrative solver based previous iterrations. That means that in order to calculate a position, the resulting force in that particular iterration needs to be calculated. The resulting force generates an acceleration which is then used to calculate how the velocity changes from the previous iterration. The new velocity then generates a new position.
The model is initiated using set values, for instance setting all accelerations and velocities to 0.

2.1.1 Thrust

First the RPM of the propeller is calculated. In order for the motor to not instantly go to the desired RPM, the time derivative of the RPM, dRPM is controlled:

$$dRPM = \begin{cases} \gamma_1 \cdot throttle, & \text{if } throttle > 0 \\ -\gamma_2, & \text{if } throttle = 0 \end{cases} \tag{2.1}$$

where $throttle = [0, 1]$ is the percentage of throttle, γ_1 is a constant depending on how fast the engine can increase the RPM and γ_2 is how fast the engine drops the RPM.

The RPM of the engine is then calculated as:

$$RPM = \begin{cases} RPM_{min}, & \text{if } RPM_{last} < RPM_{min} \\ RPM_{max}, & \text{if } \int_t^{t+\Delta t} dRPMdt > RPM_{max} \\ \int_t^{t+\Delta t} dRPMdt, & \text{else} \end{cases} \tag{2.2}$$

where, RPM_{min} is the idle RPM of the engine, RPM_{max} is the maximum RPM of the engine, RPM_{last} is the sample from the last iteration of the simulation.

With the RPM of the engine, the velocity over the propeller can be calculated from knowing the axial velocity into the propeller, which is assumed to be the same as the vessel, $\dot{x}$, and the rotational velocity of the propeller along with propeller data. The average rotational velocity of the propeller can be calculated as the rotational velocity of the gearbox divided by the gearratio. The average rotational velocity can then be calculated as the rotational velocity at the middle of the radius:

$$\Omega = \frac{r_{prop}}{2}\frac{RPM}{R}, \tag{2.3}$$

The efficiency that the propeller turns the water is simplified to:

$$\eta = (1 - slip)/\tan(90 - pitch), \tag{2.4}$$

The resulting rotational velocity in the propeller can then be calculated as $\eta\Omega$.

The total velocity in the propeller is simplified to that the propeller only turns the water. That means that the resulting velocity can be calculated:

$$v_{prop} = \sqrt{(\eta\Omega)^2 + \dot{x}^2}, \tag{2.5}$$

where r_{prop} is the radius of the propeller, R is the gearratio, $slip$ is the slip ratio of the propeller (how well the propeller grips in the water) and $pitch$ is the pitch of the propeller.

The longitudinal velocity out of the propeller can be calculated from the mass flow, $\dot{m}$, which is a function of the velocity in the propeller [12]:

$$\dot{m} = \pi r_{prop}^2 V_{prop}\rho, \tag{2.6}$$

where ρ is the density of water.

The axial velocity, C_{x2}, is then calculated from the isentropic work carried out by the propeller [12]:

$$C_{x2} = \sqrt{\left(\frac{\dot{m}\dot{x}^2}{2} + f(RPM)\right)\frac{pods}{\dot{m}}}, \tag{2.7}$$

where, $f(RPM)$ is the power delivered by the engine as a function of the engine RPM and $pods$ are the number of pods or propellers on the vessel.

The thrust can then be calculated as the change of momentum:

$$F_{thrust} = \dot{m}(C_{x2} - \dot{x}), \tag{2.8}$$

2.1.2 Drive System

The turning force of a boat depends highly on the motor system. For a conventional system, where the propellers positions are fixed relative the boat, the force comes from the rudder. On an Inboard Performance System, IPS, where the propellers are

installed on pods which rotate, the turning force comes from changing the angle of the propellers similarly to how a rudder is changed on a conventional system.

To find the thrust force in the inertial coordinate system, two different equations were used depending on if it is an IPS system or a conventional system:

$$\mathrm{F}_{throtte_x} = \begin{cases} 2\cos(\delta)\cos(p)\dot{m}(C_{x2} - \dot{x}), & \text{for } IPS \\ 2\cos(p)\dot{m}(C_{x2} - \dot{x}), & \text{for conventional} \end{cases}$$

$$F_{throtte_y} = \begin{cases} 2\sin(\delta)\cos(p)\dot{m}(C_{x2} - \dot{x}) + \frac{1}{2}\sin(\delta)A_{rudder}\rho C_L(C_{x2} - \dot{x})^2, & \text{for } IPS \\ \frac{1}{2}\sin(\delta)A_{rudder}C_L\rho(C_{x2} - \dot{x})^2, & \text{for conventional} \end{cases}$$

$$F_{throtte_z} = 2\sin(p)\dot{m}(C_{x2} - \dot{x}), \text{ for } both \tag{2.9}$$

In order to calculate the drag force from the water, the wetted reference area needs to be calculated. The cross-section of the hull is assumed to be an equilateral triangle. The height of the triangle in rest is equal to the submerged depth of the vessel while in rest, h_0, and the width of the triangle is equal to the width of the surface line in rest, B_0. Since it is assumed that the hull is an equilateral triangle the angle α can be calculated and will stay the same independent of the submerged depth (see Figure 2.2).

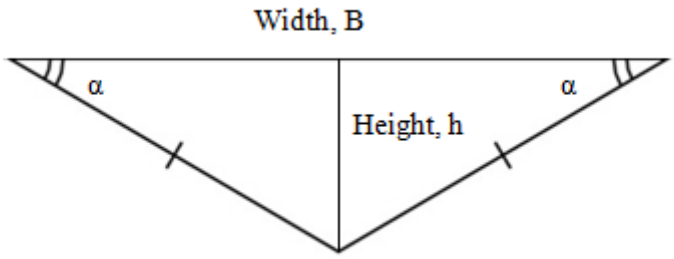

Figure 2.2: Schematic of the simplified hull.

The instantaneous height is calculated as:

$$h = \begin{cases} h_0 + z, & \text{if } h_0 + z < 0 \\ 0, & \text{else} \end{cases} \tag{2.10}$$

The instantaneous width and area is then calculated as:

$$B = 2h\tan\alpha, \; Area = \frac{Bh}{2} \tag{2.11}$$

The fluid dynamic forces can then be calculated as [12]:

$$F_{fluid_x} = \frac{1}{2}\rho\dot{x}^2 C_{D_x} Area + A_{rudder}\rho \sin\delta(Cx_2 + \dot{x})^2 \tag{2.12}$$

$$F_{fluid_y} = \frac{1}{2}\rho\dot{y}^2 C_{D_y} Area \tag{2.13}$$

$$F_{fluid_z} = \frac{1}{2}\rho\dot{z}^2 C_{D_z} Area - \frac{1}{2}\rho\dot{x}^2 C_L \tag{2.14}$$

Furthermore, the vessel also displaces different amount of water when moving in different directions. In order to compensate, it is assumed that the resulting force in each direction not only accelerates the mass of the vessel, but also a specific amount of water. The mass in each direction is specified as m_i, where i indicates the direction. The governing equations for translational movement can then be formulated:

$$\ddot{x} = \frac{F_{throtte_x} - F_{fluid_x}}{m_x} \tag{2.15}$$

$$\ddot{y} = \frac{F_{throtte_y} - F_{fluid_y}}{m_y} \tag{2.16}$$

$$\ddot{z} = \frac{F_{throtte_z} - F_{fluid_z}}{m_z} \tag{2.17}$$

The governing equations for rotational movement can be found from the relation between angular momentum and torque, $\frac{dL}{dt} = \tau$, where L is the angular momentum and τ is the torque. First the torque is calculated according to.

$$\tau = lF_{resulting}, \tag{2.18}$$

where l is the lever arm between the rotation axis and $F_{resulting}$ is the resulting force around that axis.

Using the definiton of angular momentum, the angular acceleration can then be found:

$$\tau = I\ddot{\omega}, \tag{2.19}$$

where I is the inertia around the rotational axis.

The rotational axis is assumed to be located in the Center of Bouyancy (CoB) for each rotational direction.

2.2 PID Controller

To control the throttle and rudder output a PID-controller is used. The PID-controller is chosen because it is easy to implement, relatively simple to tune and proven to be a reliable controller in most cases. PID stands for proportional–integral–derivative action. As can be understood from the name, it consists of three different parts that

are added together to create the output signal. The PID-controller is a feedback controller, which means that it feeds back the output signal and subtracts it from the desired signal to create an error. The error is manipulated in three different ways to create a new output signal [9]. The control scheme can be seen in Figure 2.3.

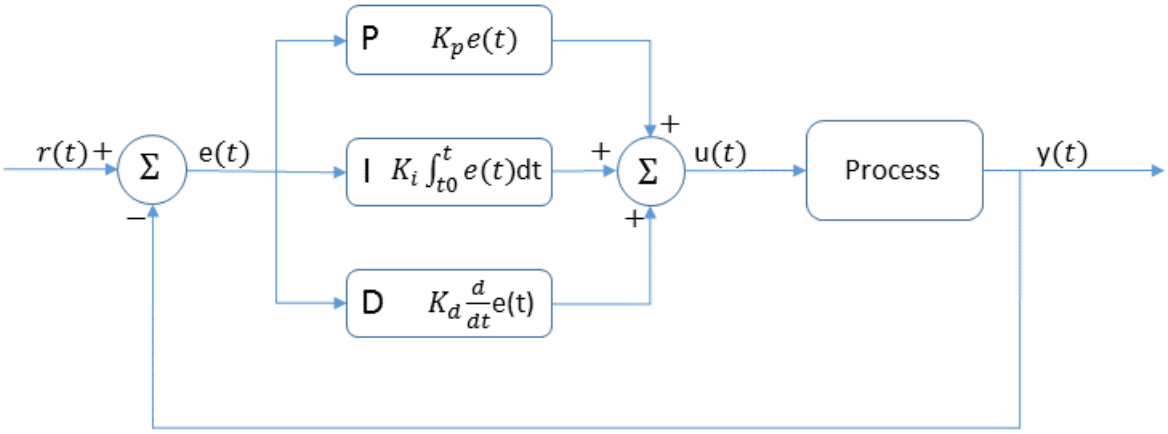

Figure 2.3: Schematic of a PID-controller

In the figure, $r(t)$ is the desired reference signal. The difference between the refrence and the actual output $y(t)$ represents the error, $e(t)$, in the system. The error is manipulated in three different ways and the signals are added then together as $u(t)$ which is the input signal to the process. The idea behind the control loop is to lower the error in each sample until an adequate output signal is reached.

The controller have three tuning parameters, K_p, K_i, and K_d, where each affects one part of the controller.
K_p is a proportional gain that produces an output signal which is proportional to the error. A large gain results in a large change in the control signal, which makes the system fast but it may cause an overshoot of the desired value, and, it can also cause problems with stability of the system. On the other hand, if the gain is not sufficently large, it cannot regulate the signal enough to attenuate the disturbances. The proportional term, P_{out} is given by 2.20

$$P_{out} = K_p e(t) \tag{2.20}$$

A common issue with a proportional controller is steady-state errors. The steady-state error is a result of the construction of the proportional controller since it needs a non-zero error to drive the controller. In general, the larger the gain of the controller and the process the smaller the steady-state gain.

K_i is the tuning parameter for the integral part of the controller. The integral part, I_{out}, is used to reduce the steady-state error. The error is accumulated over time and multiplied with the integral gain:

$$I_{out} = K_i \int_{t_0}^{t} e(t)dt \tag{2.21}$$

and then added to P_{out}. The error will continue to increase as long as there is a steady-state error. The effect is that the influence of the intergral part will increase by every iteration until the steady-state error is negligable. If the integral becomes too large however, there is a risk of the system overshooting the target value.

The influence from the derivative term, D_{out} is proportional to the rate of change of the error. It is calculated by taking the derivative of the error and multiplying it by the derivative gain K_d, i.e.

$$D_{out} = K_d \frac{de(t)}{dt} \tag{2.22}$$

The derivative of the error is the rate of change of the error and thus predicts the system behaviour. It is used to reduce the settling time and improve the stability margins of the system. Often small derivative gains are used because the sensitivity to noise in the system increases with K_d. It can also be effective to use a low pass filter on the error signal.

In the simulation environment, where all signals are perfect, it is possible to tune the regulator to an almost perfect output signal. This was done by first tuning the regulator using Ziegler-Nichols method and then manually fine tune the gains to further improve the performance.

The Ziegler Nichols method was invented by John G. Ziegler and Nathaniel B. Nichols during 1940. The method is thereby old and well tested. The method is a so called frequency method and it is known to give a responsive system with short settling times. The downside with the relatively aggressive tuning is that it has a tendency to cause large over-shoots. In order to tune the PID-controller with the Ziegler-Nichols method the following methodology is applied:

- Set K_i and K_d to zero.
- Set K_p to a low neutral value.
- Set a desired setpoint.
- Increase K_p until the output reaches a state where it has stable and consistent oscillations. This K_p is called "ultimate gain", and denoted K_u.
- Measure the oscillation period and denote it T_u.

These values of K_u and T_u are then used to calculate K_p, K_i and K_d taking into account what type of regulator that is being used. The values are set according to Table 2.1.

Table 2.1: Table over tuning parameters for Ziegler-Nichols method [8]

Ziegler-Nichols method			
Control Type	K_p	K_i	K_d
P	$0.5K_u$	–	–
PI	$0.4K_u$	$0.8T_u$	–
PID	$0.6K_u$	$0.5T_u$	$0.125T_u$

2.3 Reference Heading Model

Early in the project it was realized that some form of steering algorithm would be needed in order to test the velocity algorithm. Developing a steering algorithm was not within the scope of the project, instead an algorithm was chosen from an already exisiting algorithm [5]. The algorithm works such that it calculates the current heading from GPS data according to:

$$\mathbf{heading} = (\mathbf{X}_{current} - \mathbf{X}_{last})/|\mathbf{X}_{current} - \mathbf{X}_{last}|, \tag{2.23}$$

where $\mathbf{X}_{current}$ is the current GPS position, $\mathbf{X}_{last}$ is the GPS position in the previous sample and **heading** is the heading vector with components in x- and y-direction. Current autopilot system takes reference from degrees north. This means that 0° is north, 90° is east, 180° is south and 270° is west. In order for the heading to have the same reference, the following equation was used:

$$heading = atan2\left(\frac{\mathbf{heading}_x}{\mathbf{heading}_y}\right), \tag{2.24}$$

where *atan2* is an arctangent function which takes into account what quadrant the components are in.

2.4 Heading Algorithm

Two different forms of heading algorithms were tested. The first one, called *pure pursuit*, is a very simple algorithm. In order to generate a more stable steering system a *look-ahead algorithm* was also developed [5].

2.4.1 Pure Pursuit Algorithm

The *pure pursuit algorithm* basically calculates the direction vector between the current GPS position and the next waypoint and thus the heading. The advantage of this algorithm is that it is very simple, but on the other hand it generates a very unstable steering.

In order to calculate the heading, the current GPS position and the next waypoint's GPS position are used:

$$heading = atan2\left(\frac{(\mathbf{X}_{current} - \mathbf{X}_i)_x}{|\mathbf{X}_{current} - \mathbf{X}_i|_y}\right), \tag{2.25}$$

where X_i is the waypoint.

2.4.2 Look-Ahead Algorithm

The *look-ahead algorithm* works such as that it always looks r meters in front of the vessel and finds its heading from there. That means that if you consider a circle of r meters around the vessel and a line between the boat and the next waypoint, then the intersection between the line and the circle is the reference point,as illustrated in Figure 2.4.

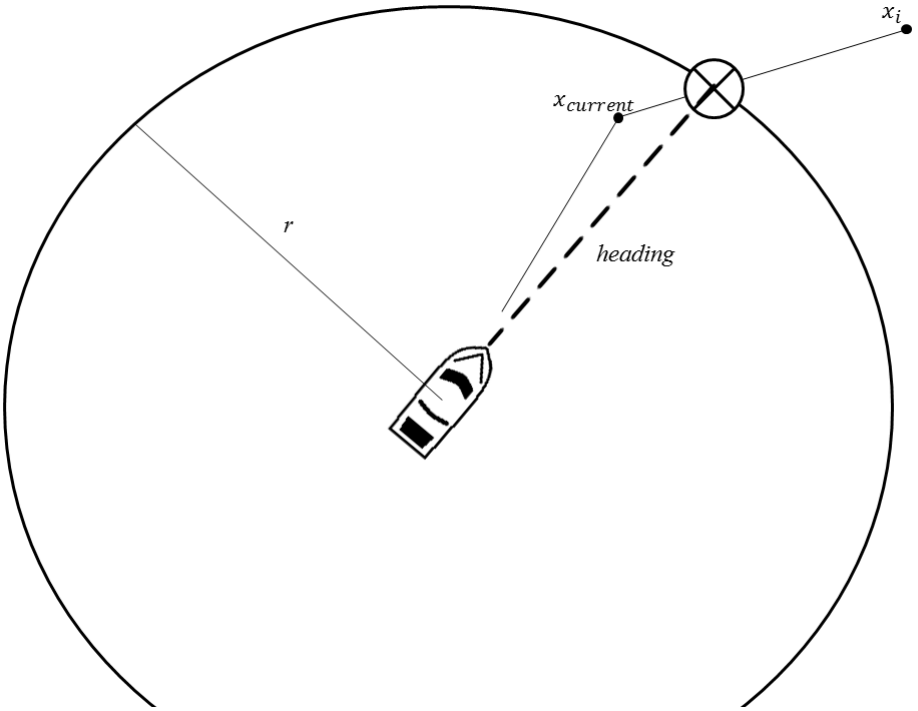

Figure 2.4: Overview of *look-ahead algorithm*.

The idea is that the reference heading will not change drastically when closing in on the next waypoint. In the *pure pursuit algorithm*, *heading* will change drastically if the vessel is not approaching the waypoint perfectly but instead slightly misses it. In Equation 2.25 both the nominator and denominator approaches zero as the vessel approaches the waypoint, which yields for very volatile changes in the values of *heading*. The equation for the *look-ahead algorithm* calculates the intersection between the circle of radius r around the vessel and the path. First the line between the current waypoint and the next is calculated:

$$y_{intersection} = kx + m$$

$$k = \frac{y_{current} - y_i}{x_{current} - x_i}$$

$$m = y_i - kx_i \tag{2.26}$$

The intersection is then found by solving:

$$r = \sqrt{(x_{vessel} - x_{intersection})^2 + (y_{vessel} - y_{intersection})^2} \tag{2.27}$$

where, x_{vessel} is the position of the vessel and $x_{intersection}$ is the position of the intersection.

The heading is then:

$$\mathbf{heading}_{intersection} = atan2\left(\frac{(\mathbf{X}_{intersection} - \mathbf{X}_{vessel})_x}{|\mathbf{X}_{intersection} - \mathbf{X}_{vessel}|_y}\right), \tag{2.28}$$

where $\mathbf{heading}_{intersection}$ is the heading towards the next intersection.

2.5 Reference Velocity Model

The reference value for the velocity algorithm has also been divided into two different algorithms. One which focuses on the comfort in the vessel and one which focuses on the path following ability of the vessel.

2.5.1 Comfort

The comfort in a vessel can be expressed in terms of true turn [11]. True turn is a measurement between the difference in resulting acceleration and the roll of the vessel. The resulting acceleration is the resulting vector between the lateral acceleration and the gravity. The lateral acceleration always has the direction outwards in a turn while the gravity always has the direction downwards. From this the resulting acceleration can be calculated: $\beta = \text{atan}\,(a_{lat}/g)$, where a_{lat} is the lateral acceleration measured in g's where g is the gravitational constant. The difference between the β and the roll angle is then difference from true turn. That is if the roll angle and β are the same, then it is 100% true turn.

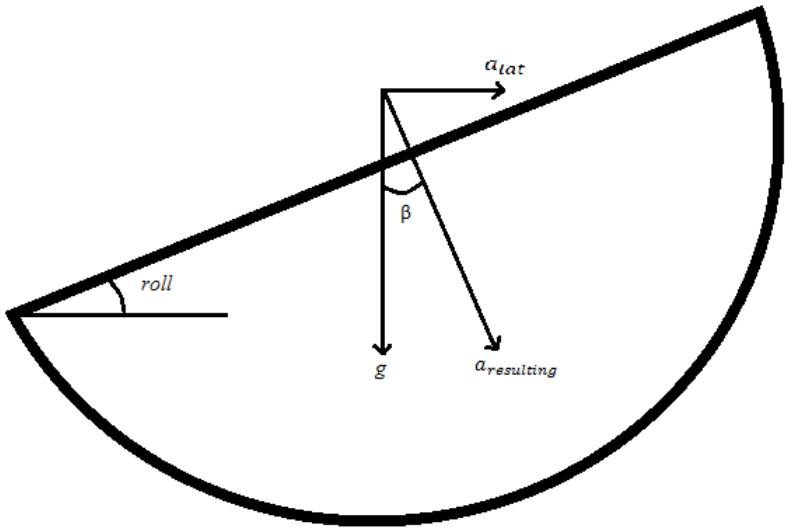

Figure 2.5: Illustration of true turn

In Figure 2.5 true turn is visualized. For a true turn, a person in the vessel will not be forced sideways since the resulting acceleration will be perpendicular to the floor of the vessel, and thus will not experience any difference except the perceived increased gravitation. The level of comfort is often categorized as in Table 2.2

Table 2.2: Table over different levels of comfort for a turning boat

Angle deviant from true turn	
$0-5°$	Comfortable
$5-10°$	Feels unnatural
$10-15°$	Not comfortable
$15° - above$	Unpleasant

In order to find the reference velocity for a true turn, the lateral acceleration and the roll angle are measured via an IMU. For a true turn the roll angle and the $a_{lat_{resulting}}$ are equal:

$$roll = a_{lat_{resulting}} = \text{atan}\left(a_{lat_{preferred}}/g\right) \implies a_{lat_{preferred}} = \tan(roll) \cdot g. \tag{2.29}$$

For a turn, the lateral acceleration is equal to the centripetal acceleration, a_c:

$$a_c = \frac{v^2}{r}, \tag{2.30}$$

where v^2 is the velocity of the vessel and r is the radius of the circle. The current radius of a turn can be calculated from the actual lateral acceleration, $a_{lat_{actual}}$:

$$r = \frac{v^2}{a_{lat_{actual}}}. \tag{2.31}$$

That means the preferred velocity can be calculated as:

$$v_{preferred} = \sqrt{a_{lat_{preferred}} \cdot r} = \sqrt{\tan(roll) \cdot g \cdot \frac{v^2}{a_{lat_{actual}}}}, \tag{2.32}$$

One big downside to this system, however, is that it is purely reactional. All the values are taken instantaneously, and since a marine vessel is a very delayed system between user input and reaction, the resulting output can be off.

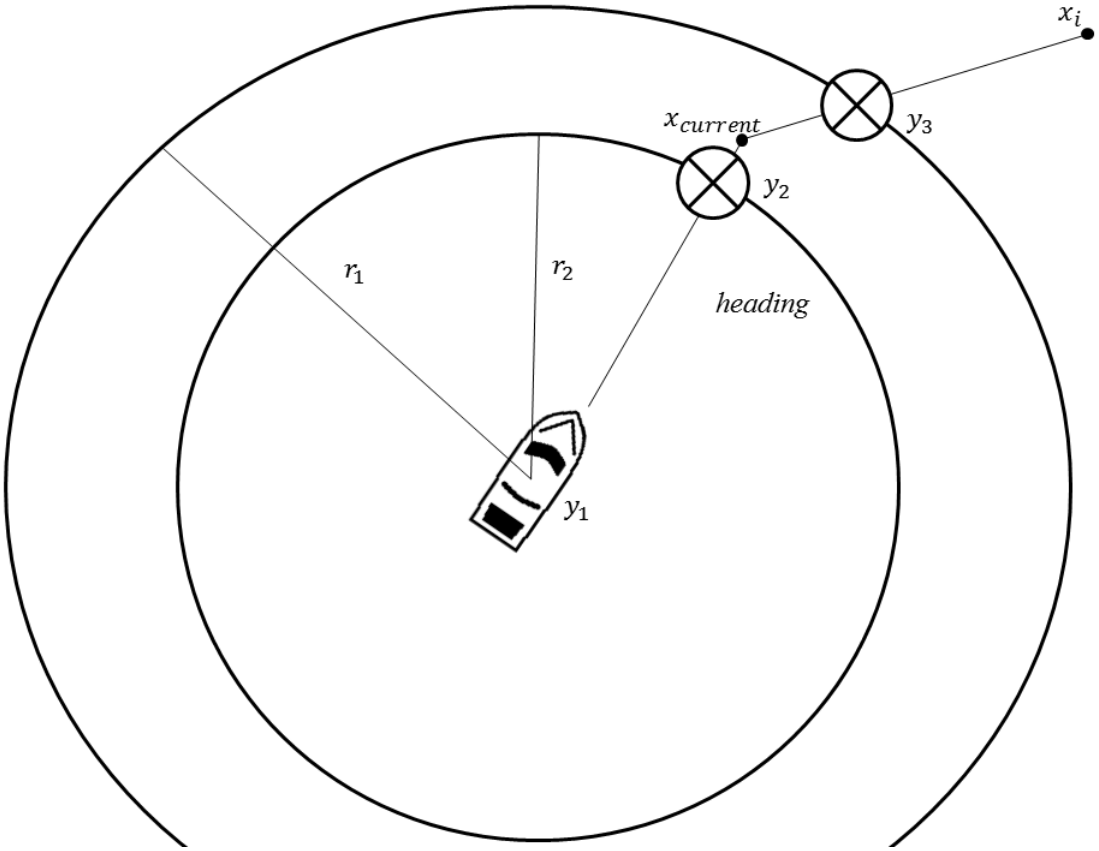

Figure 2.6: Method of finding instantaneous radius. Two different look-ahead distances are chosen, r_1, r_2 and from the intersections y_1, y_2, y_3 the radius of the corner is calculated.

In order to generate a predictive system, the system above, the radius and the roll angle need to be predicted. The radius can be predicted since the route is known, however, the roll angle is more difficult to predict. In order to create a predictive algorithm, only the lateral acceleration is taken into account. That means instead of calculating a preferred lateral acceleration as in Equation (2.29), a set value of preferred lateral acceleration can be set. The radius of a turn can be calculated from three points (y_1, y_2, y_3 in Figure 2.6). In order to find these three points, a

similar algorithm to *look-ahead algorithm* was developed. Instead of using one look-ahead parameter, two are used. From these two distances two intersection points between the path and the vessel are calculated in the same way as in the *look-ahead algorithm*. From these three points the radius of the circle they form is calculated:

$$r_{turn} = \frac{l_{12}l_{23}l_{31}}{4A} \tag{2.33}$$

where l_{ij} is the distance between point y_i and y_j and A is the area of the triangle they form. It is assumed that the circle of the three points match the curvature of the upcoming turn (see Figure 2.7).

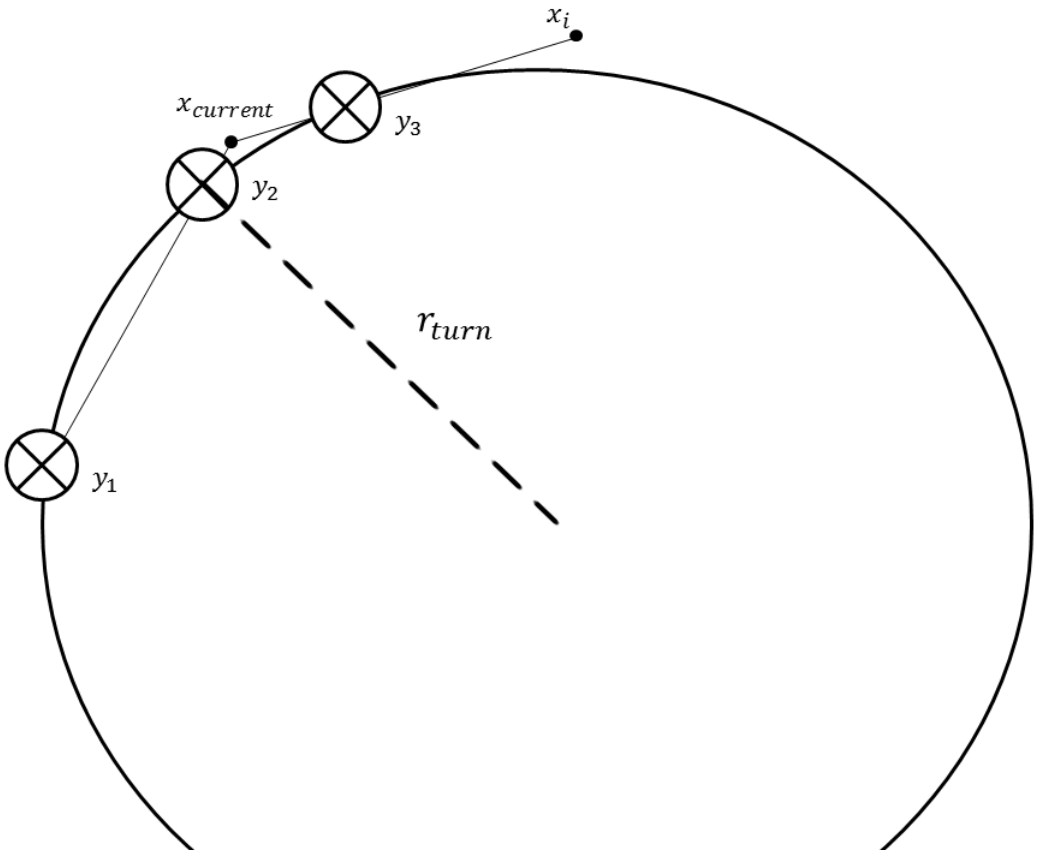

Figure 2.7: Radius of a turn calculated from three points.

The predicted preferred velocity can be calculated as:

$$v_{preferred} = \sqrt{a_{lat_{preferred}} \cdot r_{turn}}, \tag{2.34}$$

where $a_{lat_{prefered}}$ is a set constant and r_{turn} as in Equation (2.33).

2.5.2 Path Following

The mean radius of a turn can be calculated using a free body diagram of the vessel. Assume steady-state corner, no acceleration in the x-direction and, small body slip agnles and steering angles.

From the assumptions we can see that the only existing acceleration is turning the vessel, which means:

$$\ddot{\omega} = 0 \tag{2.35}$$

The resulting torque around the vessels own rotational axis can be computed by:

$$T = l(F_{throtty} - CD_A(1/2length\dot{\omega})^2), \tag{2.36}$$

where l is the length of lever arm between the center of rotation and the applied thrust and $length$ is the length of the vessel.
From the angular momentum, the rotational acceleration is then found:

$$T = \frac{dL}{dt} = \frac{dI_{zz}\dot{\omega}}{dt} = I_{zz}\ddot{\omega} = l(F_{throtty} - CD_A(1/2length\dot{\omega})^2)$$

$$\Longrightarrow$$

$$\ddot{\omega} = \frac{l(F_{throtty} - CD_A(1/2length\dot{\omega})^2)}{I_{zz}} = 0 \tag{2.37}$$

Equation (2.37) can only hold if $F_{throtty} = CD_A(1/2length\dot{\omega})^2$, which means that:

$$\dot{\omega} = \sqrt{\frac{F_{throtty}}{CD_A}}\frac{2}{length}. \tag{2.38}$$

Furthermore, the mean radius of a turn can be calculated as $R = \frac{v}{\dot{\omega}}$. Inserting Equation (2.38) the radius can be calculated as:

$$R = \frac{v}{\sqrt{\frac{F_{throtty}}{CD_A}}\frac{2}{length}} = \frac{v \cdot length\sqrt{CD_A}}{2\sqrt{F_{throtty}}} = C\frac{v}{\sqrt{F_{throtty}}} \tag{2.39}$$

where $C = length\sqrt{CD_A}/2$ is a specific constant for the vessel. The constant can be found from sea trials, the velocity is known from sensors and the thrust from the engine can be found as a function of the throttle from sea trial. The sea trials would then involve to first go in a straight line with constant throttle until steady-state is achieved, and then rapidly going to 0% throttle to find the thrust from the engine from Newton's second law, while the constant C can be found by solving the equation for a known turn.

2.5.3 Constant Lateral Acceleration

The second path following algorithm is very similar to Equation (2.32), with the difference that the look-ahead distance is dynamic. For a given set lateral acceleration, the force in the y-direction is $F_y = ma = CD_A\ddot{y}$. If the lateral acceleration is constant, the left hand side of the equation is constant, the velocity the vessel travels sideways with is constant, assuming that the drag coefficient and the reference area stays constant. The allowed lateral velocity can then be found by calibration and

set to the needed path following ability.

In order for the regulator to be able to slow down in time to reach the desired lateral acceleration, the algorithm uses a dynamic look-ahead distance. Instead of finding the instantaneous radius, the radius at $l(v)$ ahead of the vessel is used, where $l(v)$ is a length depending on the velocity (see Figure 2.8). The function $l(v)$ can be set in different ways depending on the vessel. One simple way of using it is simply setting it proportional to the velocity.

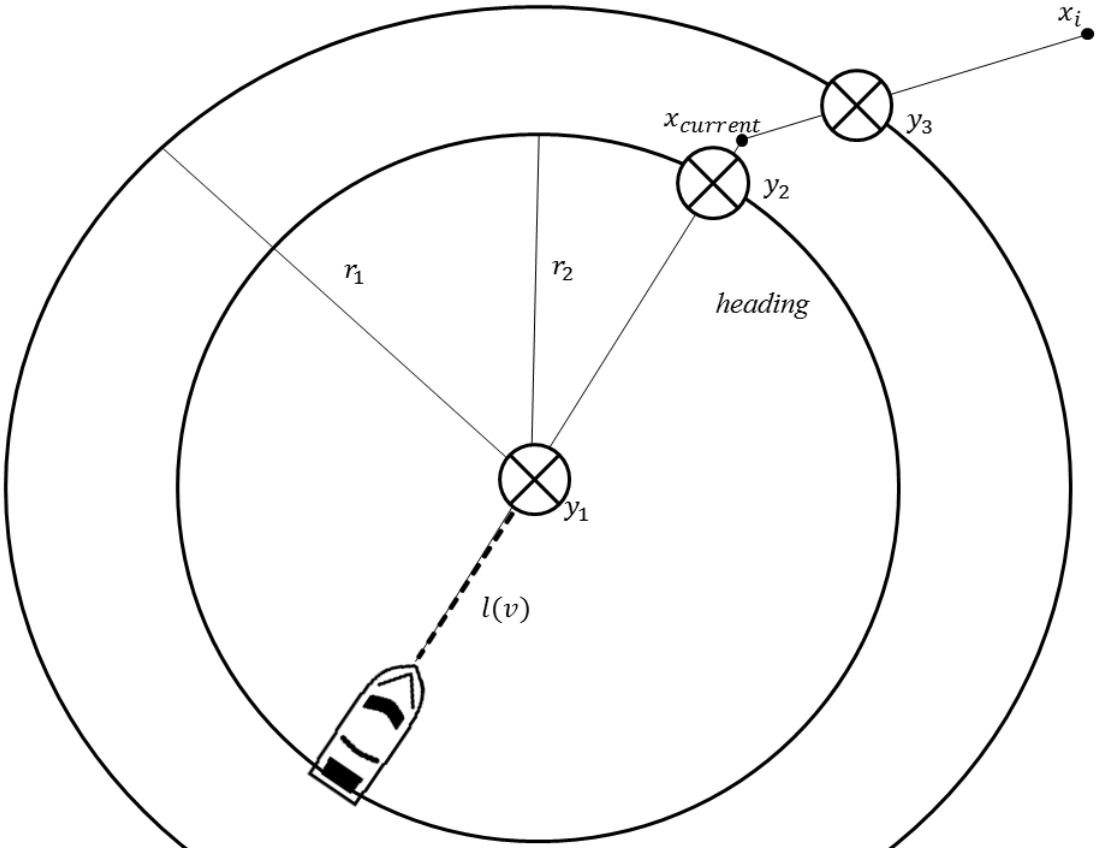

Figure 2.8: Illustartion of how the radius of the corner $l(v)$ meters in front of the vessel is calculated. The three points, y_1, y_2 and y_3 are then used to calculated to radius of the turn as illustrated in Figure 2.7

2.5.4 Route Optimization

In addition to regulation of the throttle an algorithm to smoothen the path was created. The idea is to use cubic interpolation between the points in order to create additional waypoints that give a more continuous route to follow [5].

The parameterization of the GPS coordinates of the waypoints, X, Y, are created with $t \in [0, 1]$ as:

$$x(t) = a_1 + b_1 t + c_1 t^2 + d_1 t^3 \tag{2.40}$$

$$y(t) = a_2 + b_2 t + c_2 t^2 + d_2 t^3, \tag{2.41}$$

where a_i and b_i are parameterization constants.
In order to achieve a smooth transition between waypoints, the derivative in each waypoint needs to be calculated. The derivative can be chosen in different ways, the following method is suitable for marine autopilot applications [5]:

$$X_i' = \frac{||X_{i+1} - X_i||(||X_{i+1} - X_i|| \left(\frac{X_{i-1} - X_{i-2}}{2}\right) + ||X_i - X_{i-1}|| \left(\frac{X_i - X_{i-1}}{2}\right))}{||X_{i+1} - X_{i-1}||} \tag{2.42}$$

$$Y_i' = \frac{||Y_{i+1} - Y_i||(||Y_{i+1} - Y_i|| \left(\frac{Y_{i-1} - Y_{i-2}}{2}\right) + ||Y_i - Y_{i-1}|| \left(\frac{Y_i - Y_{i-1}}{2}\right))}{||Y_{i+1} - Y_{i-1}||} \tag{2.43}$$

That means that the boundary conditions for Equation (2.40) and (2.41) are:

$$\begin{array}{rcl} x(0) & = & X_i \\ x(1) & = & X_{i+1} \\ y(0) & = & Y_i \\ y(1) & = & Y_{i+1} \\ x'(0) & = & X_i' \\ x'(1) & = & X_{i+1}' \\ y'(0) & = & Y_i' \\ y'(1) & = & Y_{i+1}' \end{array}$$

3
Method

The implementation of the derived method consisted of two main parts, simulations and testing. The majority of the time was spent on creating and testing the simulation environment, while testing the algorithm in a rig as well as in a boat was done to gather data and validate the results of the simulations.

3.1 Simulation Enviroment

The simulation environment was implemented in Matlab/Simulink. The governing equations of motion derived in Section 2.1 are solved in real time, meaning that real-time simulations could be performed. This was used to visualize the vessel. Different vessels were implemented so that only certain parameters needed to be changed, such as mass, inertia etc. The idea was to fully separate the simulation environment from the autopilot, and only send sensor data, such as GPS, roll, pitch and yaw rate from the simulation model to the autopilot, to mimic sensor data, while input data to the governing equations of motion are only the throttle and the steering angle.

The physics engine and graphics engine are working in real time. Due to this, the simple and fast Euler forward solver was chosen. The use of an explicit method allows to solve the problem in real time [6].

3.1.1 Vessel Modelling

Two different vessels have been modeled, one traditional with a rudder and one with an IPS system. In order to find the necessary data to generate adequate vessel models for the physics engine, a combination of log data, data from manufacturer and data from other vessels were used. Log data could be gathered from sea trials, but it was often hard to find all the necessary data due to the nature of the testing. For instance, finding the center of gravity and the center of buoyancy is very hard from a sea trial, whereas finding the acceleration time or degrees of roll for a certain turn is simple. In order to find additional data, Volvo Penta was contacted and assisted with a CAD model of the hull of a boat of interest. From the CAD model, a simple CFD analysis was made to find the drag coefficient and lift coefficient of the hull. The CFD case was heavily simplified due to the complexity of the problem. A multiphase flow case was setup assuming that the air and water hit the boat with the same velocity. To find the missing data, values for a Viknes 830 boat was used.

It was then assumed that these values could be scaled to the size of need [7].

The accuracy for a certain vessel was not very high of interest, since the system was intended to be working on different types of boats. What was important was that the parameterization was reasonable.

3.1.1.1 Rudder

The rudder steered boat was parameterized from log data and scaling of parameters from a Viknes 830. First a length scale and a mass scale was determined, these were then used to scale the remaining parameters of the Viknes boat. It was assumed to be adequate close enough for it to be a sufficiently good starting point. These parameters were then fine tuned to fit the logged data.

3.1.1.2 IPS

The IPS steered boat had other data available. Motor data as well as a CAD of the hull was available while log data was not. This ment that fluid dynamic data could be found from CFD and the motor parameters could be found from the motor data. The mass and inertia was assumed to be similar to that of the rudder steered boat.

3.1.2 Graphics Engine

The chosen graphic engine is a game engine called Unity. Unity was chosen becuase it is a simple freeware that is made for people to learn how to build graphics for games in an easy way. Unity also has a big community where it is easy to get help if needed and also to find complete code for different features.

The graphical representation of the underlying physics engine is written in Matlab and Simulink, which sends the position data of all six degrees of freedom to Unity locally via an UDP connection in real time. Unity then interprets the signal and visualizes the position of the boat.

The environment used for the graphical representation is an actual place in Gothenburg's archipelago around Krossholmen, Brännö and Grötö, shown in Figure 3.1. The place was chosen because the initial testing of the enhanced autopilot will take place in this area. By using a known real place a route could be constructed using an arbitrary GPS plotter system, which has the possibility to extract the coordinates to an .xml file. These GPS-coordinates are then loaded into the Simulink autopilot in order to generate a route to follow and also into Unity to generate a graphical representation of the waypoints.

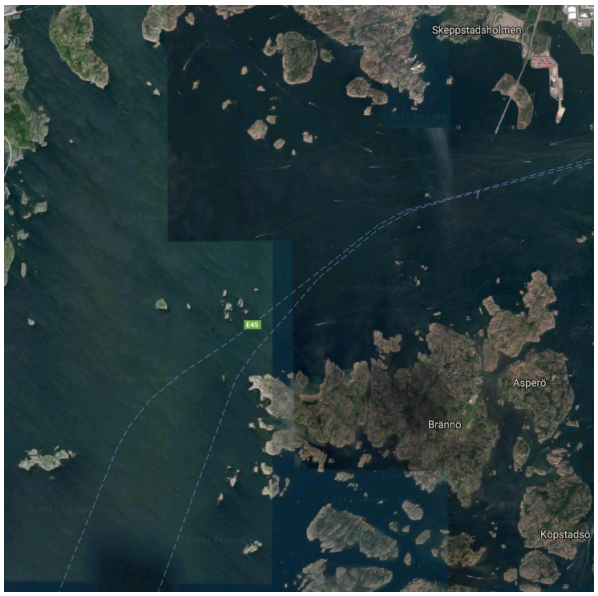

Figure 3.1: Satellite image over the actual place that is later used in the simulator.

In order to build the terrain in Unity with respect to the real environment shown in Figure 3.1, a height map from a satellite image was used. A height map is an image, where every different nuance of the colour represents a specific height. The minimum height is represented by black and maximum height is represented by white colour, see Figure 3.2.

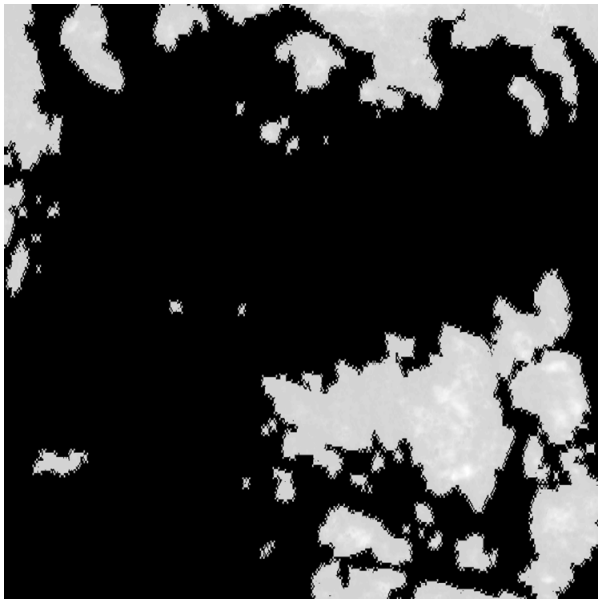

Figure 3.2: Height map over the same area as in figure 3.1 above. Image taken from © OpenStreetMap contributors [10].

This height map is then interpreted in Unity as a terrain and displayed graphically as Figure 3.3 shows.

Figure 3.3: The terrain from the height map as it looks in Unity.

To ease up the transfer and comparability between the simulator and real life it is important that it is possible to drive the same route in both applications. To do this a route was, as mentioned before, plotted in a GPS plotter system and then the coordinates were extracted to an xml document. After extracting the GPS coordinates, which are displayed in latitude and longitude decimal coordinates, they were converted into a cartesian coordinate system to fit in Unity which also uses a cartesian coordinate system.

After converting the GPS coordinates they were also re-mapped to fit the pixel space in the environment build up in Unity. Together with the known scale factor used to create the terrain in Unity it is now possible to calculate distances in meter.

3.2 Autopilot

The autopilot is implemented in two different independent algorithms, the steering algorithm and the velocity algorithm. The two autopilot algorithms have the same structure. From each model a reference value is found, which is sent to the PID regulator, which controls either the throttle or the rudder angle.

3.2.1 Steering

The reference value is calculated as in Section 2.4. The reference value algorithm uses different amount of inputs depending on the chosen algorithm. The *pure pursuit algorithm* only needs the current GPS position, the GPS position of the waypoints, and the minimum distance to the waypoint (see Figure 3.4), while the *look-ahead algorithm* also requires the look-ahead distance (see Figure 3.5).

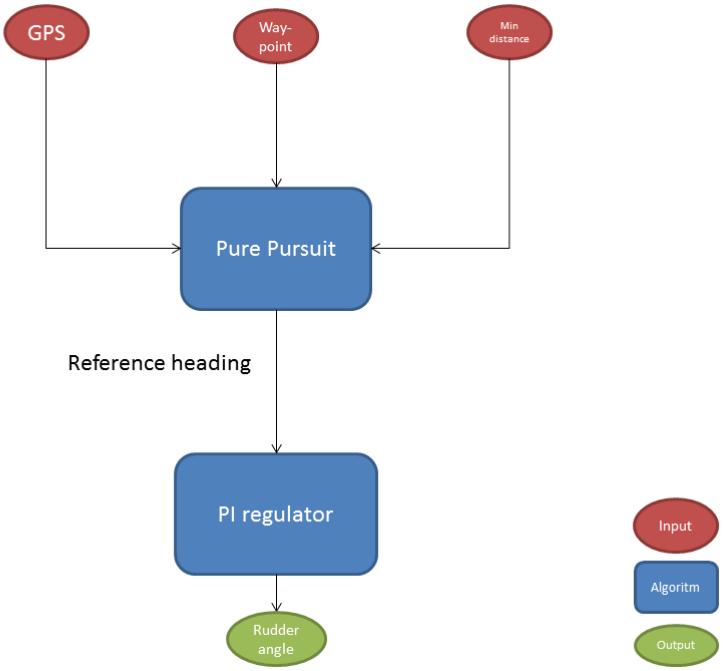

Figure 3.4: Flow chart of the *pure pursuit algorithm*.

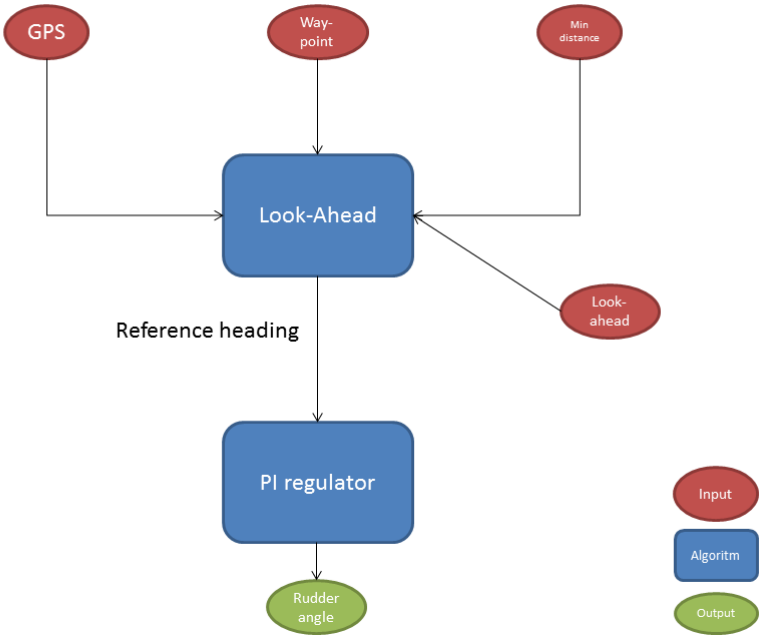

Figure 3.5: Flow chart of the *look-ahead algorithm*.

3.2.2 Velocity

The velocity algorithm is set up in the same way as the steering algorithm. The different velocity reference algorithms use different inputs. The comfort model uses roll, velocity, yaw rate and constant values of the different lateral acceleration, while the path following algorithm also uses GPS, and additional tuning parameters, such as parameters for the $l(v)$ function.

The different algorithms for comfort can be seen in Section **??** and the entire algorithm for comfort can be seen in Figure 3.6. The algorithms for the path following can be seen in Section 2.5.2 and the entire algorithm can be seen in
Figure 3.7.

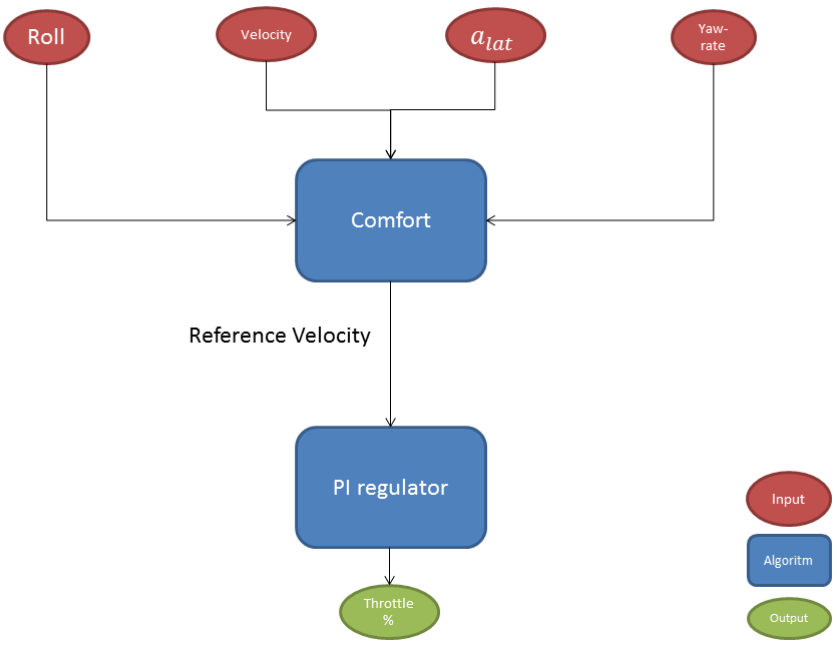

Figure 3.6: Flow chart of the comfort algorithm.

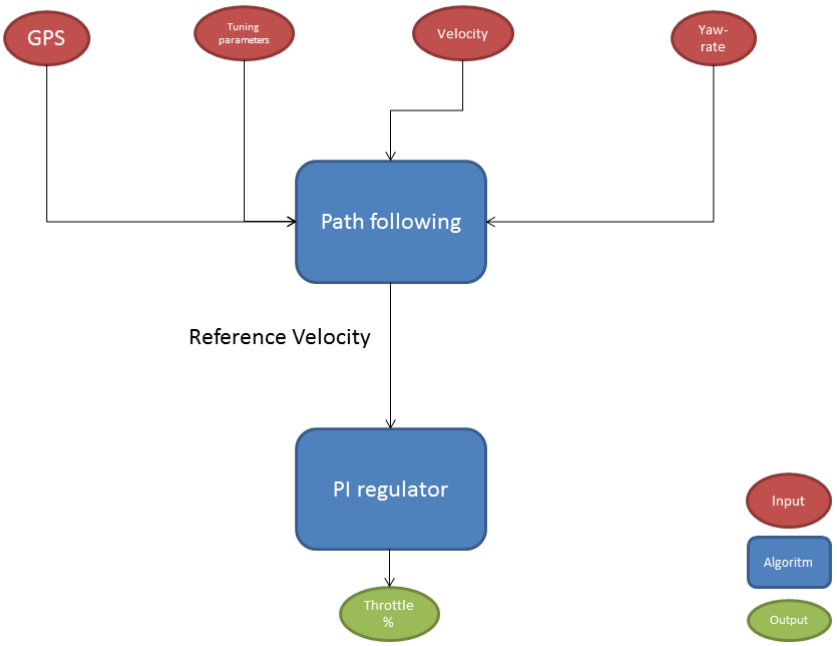

Figure 3.7: Flow chart of the path following algorithm.

3.3 Testing and Validation

When testing the autopilot two different routes are used, one route with varying cornering radius both to the port and starbord to test the cornering ability of the autopilot, and one route that is more "normal" in the sense that it mimics a potential route one could wish to drive while taking the boat out on a tour for a day. The starting point in the simulation is placed away from the first waypoint and in line with waypoint number two to give the autopilot some time to steer into the correct course and not have to make a major turn to reach the next waypoint on the route.

The autopilot is tested both with only a few waypoints on the route as for a planned route in a plotter system, but also with interpolated waypoints between the manually set waypoints. The difference of these routes can be seen in Figures 3.8 and 3.9.

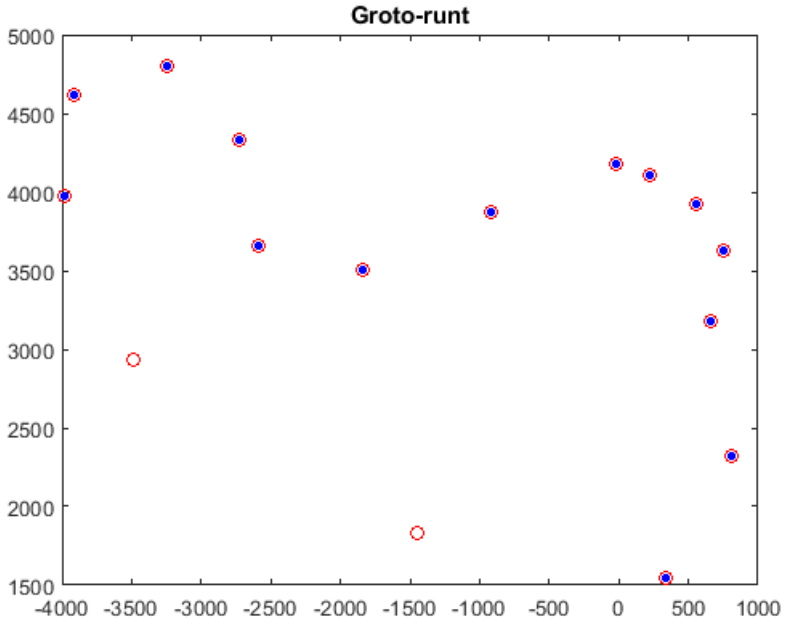

Figure 3.8: Route with manually plotted waypoints.

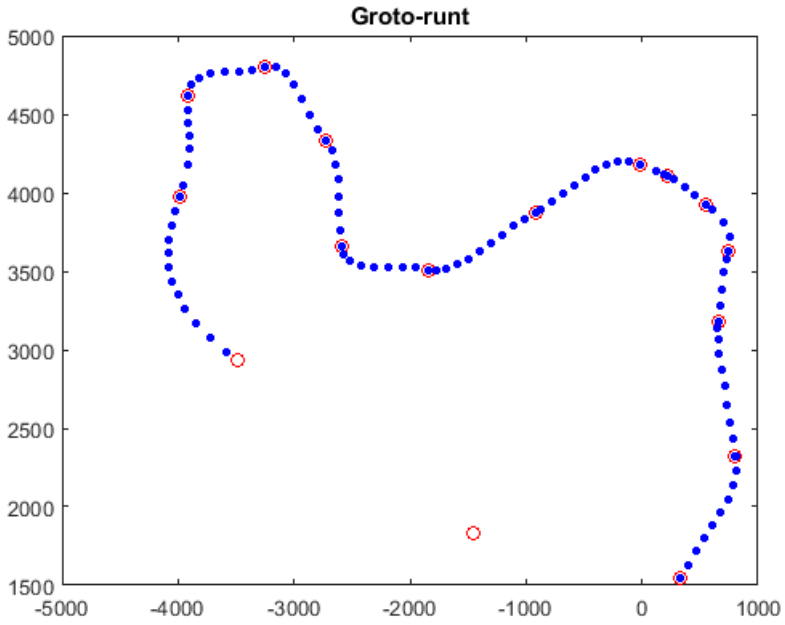

Figure 3.9: Manually set waypoints marked with a red circle and interpolated points marked with blue dots in between.

The route with interpolated points has more points for the autopilot to aim for, which should help smoothen the driven path.

In order to make the comparison between the enhanced autopilot and the standard autopilot as fair as possible the enhanced autopilot will first make a run to set

the time to finish the whole route and then the throttle is matched on the conventional autopilot to make the boat travel the same distance in the same time.

While checking log data from a GPS antenna from a boat standing still it could be noticed that the position was wandering about with a radius of roughly $50cm$. In a simulation environment there is normally no such noise. To take this noise into consideration in the simulations it is added on top of the GPS signal used in the reference velocity calculations. Instead of trying to implement waves, a static force will be added to represent a current in the sea which will force the autopilot to always compensate steering and throttle in one or another direction.

4

Results

In this chapter the results from both simulation runs and real life testing is presented. The results will not be directly comparable with each other since some simplifications are made in the simulation environment. Also, the boat models are not entirely accurate because some data and boat parameters were missing and had to be approximated using derivations from similar vessels and educated guesses. Furthermore, the tested boat and the simulated one are not the same.

4.1 Simulation

Since the influence of waves was neglected in the simulation environment a comparison between the results from the simulation environment and real life testing are expected to differ, but the results in the simulation environment are still assumed to be representative. The three different velocity algorithms were tested, and the *Constant Lateral Acceleration* algorithm was chosen to be used throughout the remainder of the tests. The autopilot was also tested both with absolute calm water and with a static force that mimics wind or current. The test runs were carried out on two different routes. One that imitates a "normal" route while the other one is more to test the autopilot's cornering ability.

In Figure 4.1 the different velocity algorithms are compared. The reference lateral acceleration is set to 0.5 in all the algorithms. The steering algorithm used is the *look-ahead algorithm*, and this steering algorithm was also used to produce the results in the following graphs.

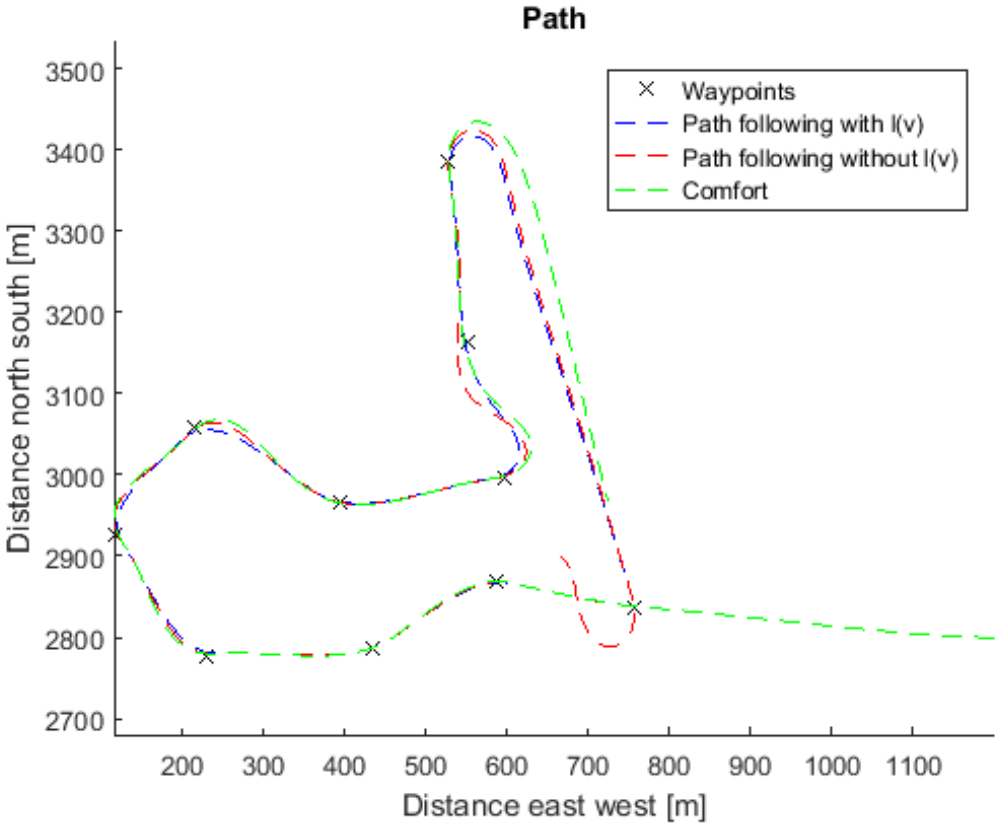

Figure 4.1: Comparison between different velocity algorithms, using the *look-ahead* steering algorithm.

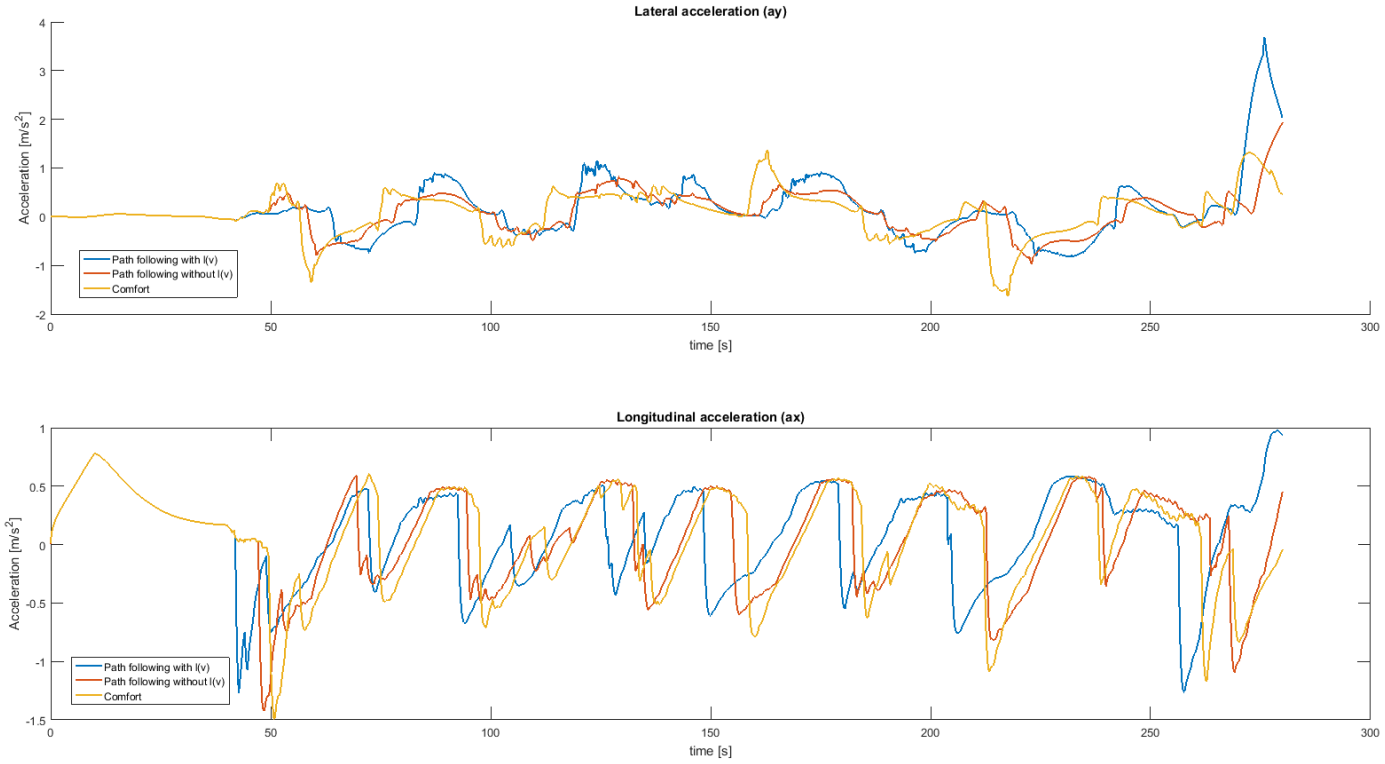

Figure 4.2: Lateral and longitudinal accelerations for different velocity algorithms.

In Figure 4.2 the different accelerations can be seen for the different velocity algorithms. It can be clearly seen that the *look-ahead algorithm* which uses $l(v)$ regulates the throttle significantly earlier than the two purely reactionary.

4.1.1 Steering algorithms

In Figure 4.3 the two different steering algorithms are shown. The throttle used is constant throttle at 50%. As mentioned earlier, the *look-ahead algorithm* is used in the following.

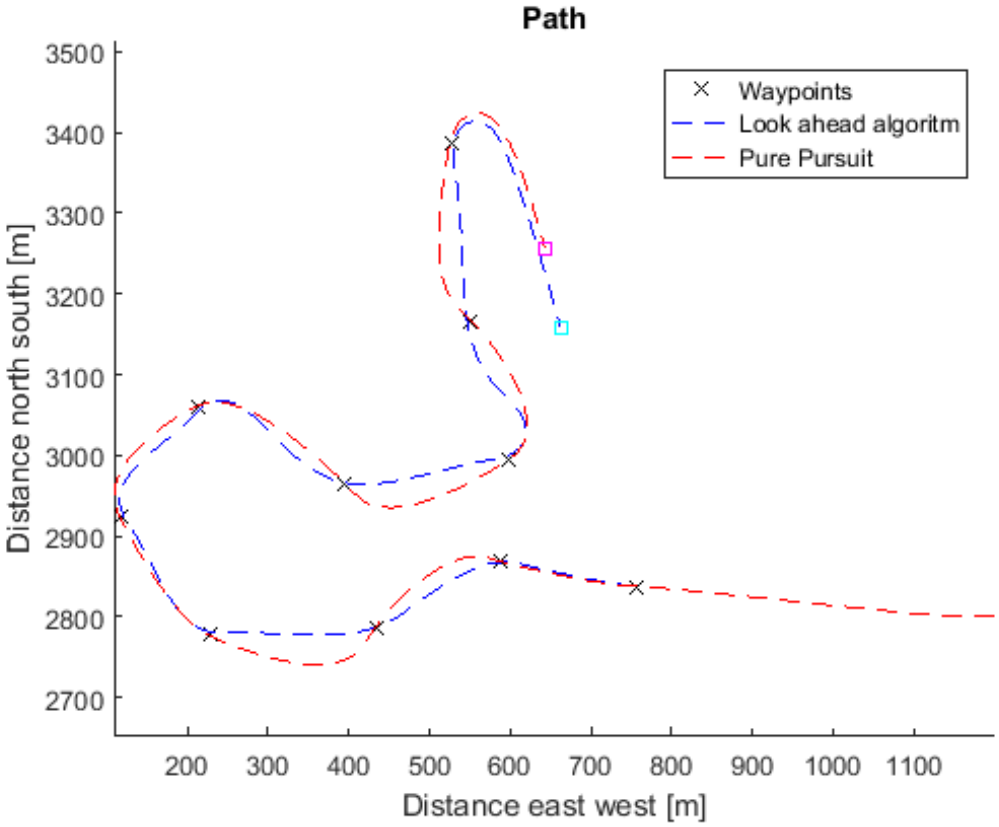

Figure 4.3: Comparison between different steering algorithms.

4.1.2 Path Following Ability

The path following ability was compared between the constant and the variable throttle on the two different test tracks. One of the larger differences between the runs with constant and variable throttle is the longitudinal speed for the curvy route (see Figures 4.5 and 4.7). Figures 4.9 and 4.11 are "normal" routes and have a much larger difference between the slower parts and the faster parts with a variable throttle. This also means that the longitudinal acceleration will vary more with a variable throttle.

As can be seen in the following graphs (Figure 4.4 and Figure 4.6) the difference between the enhanced autopilot and the conventional autopilot is not significant when the speed is low enough for the conventional autopilot to follow the route.

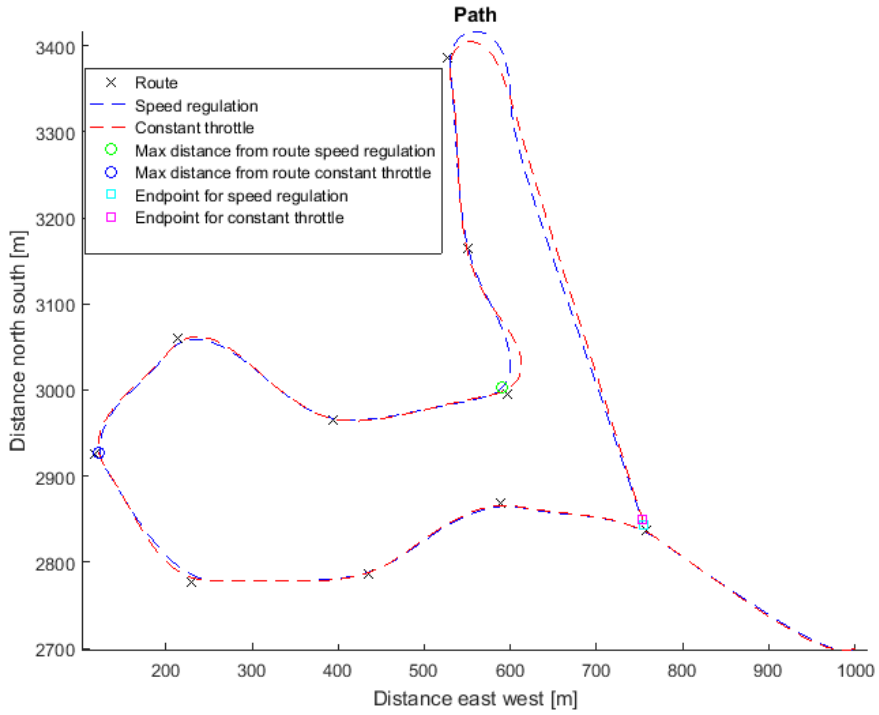

Figure 4.4: Curvy route with static current. Maximum route deviation with variable throttle is 9.59 [m]. Average route deviation with variable throttle is 4.54 [m]. Maximum route deviation with constant throttle is 5.48 [m]. Average route deviation with constant throttle is 3.37 [m].

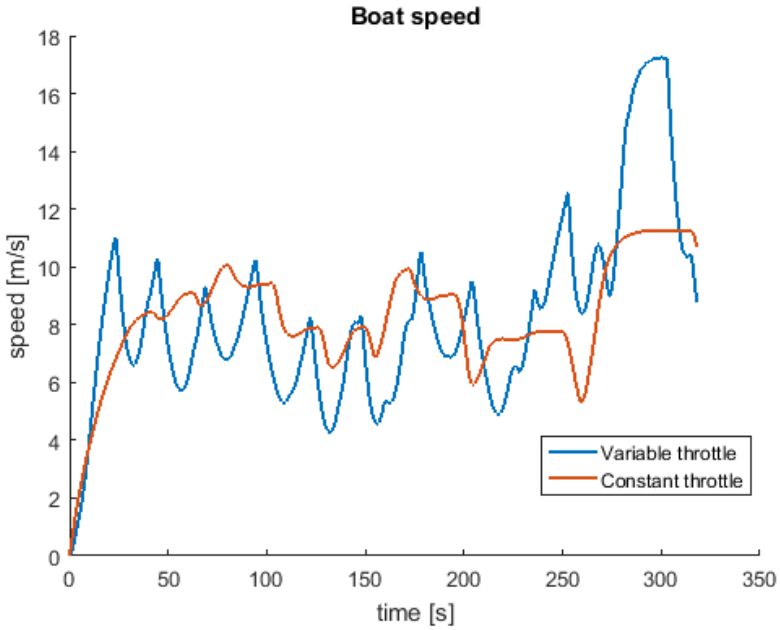

Figure 4.5: Comparison of longitudinal velocity between variable and constant throttle on the curvy route with static current.

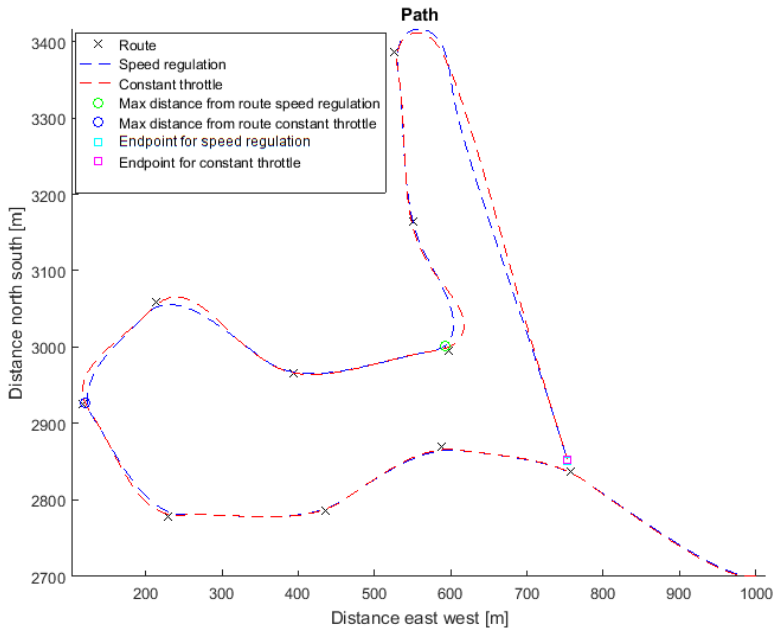

Figure 4.6: Curvy route without current. Maximum route deviation with variable throttle is 7.66 [m]. Average route deviation with variable throttle is 4.50 [m]. Maximum route deviation with constant throttle is 4.08[m]. Average route deviation with constant throttle is 2.92 [m].

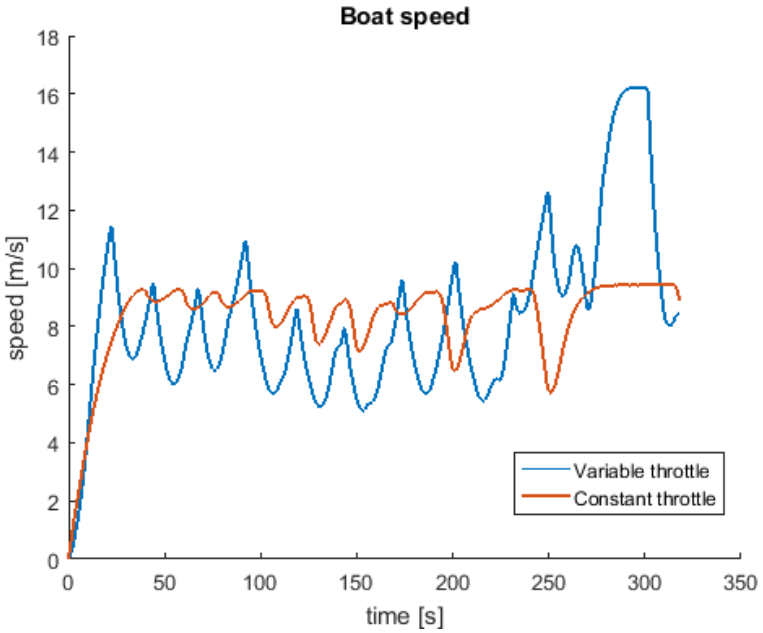

Figure 4.7: Comparison of longitudinal velocity between variable and constant throttle on the curvy route without static current.

As can be seen in Figure 4.4 for the curvy route, both autopilots manage equally well to compensate for the additional current added to the run. If the autopilots are

compared on a route with less sharp corners instead, which would feel more natural to drive, it is also noticeable that the difference between them in pure numbers becomes very small. An overview of this can be seen in Figure 4.8 below, which is with an added current, and in Figure 4.10 without the added current.

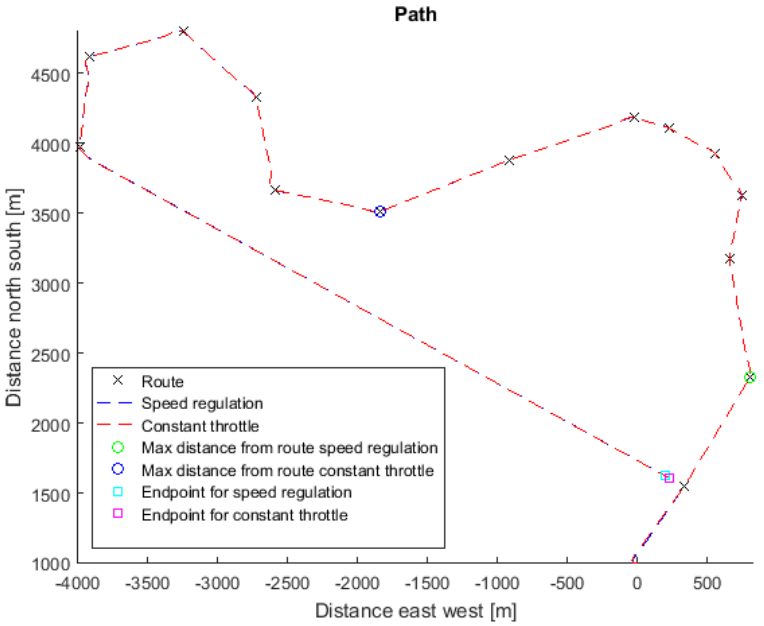

Figure 4.8: Normal route with static current. Maximum route deviation with variable throttle is 2.44 [m]. Average route deviation with variable throttle is 1.37 [m]. Maximum route deviation with constant throttle is 2.10[m]. Average route deviation with constant throttle is 1.09 [m].

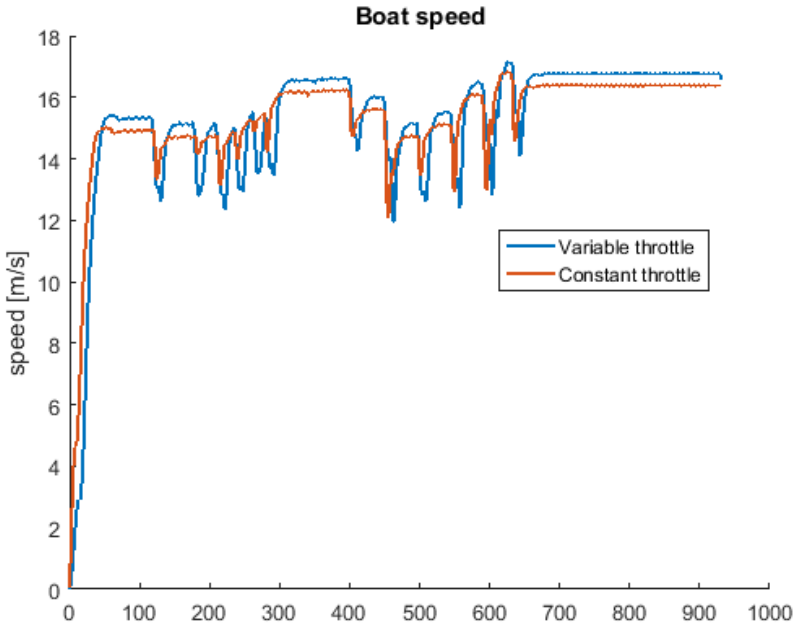

Figure 4.9: Comparison of longitudinal velocity between variable and constant throttle on the normal route with static current.

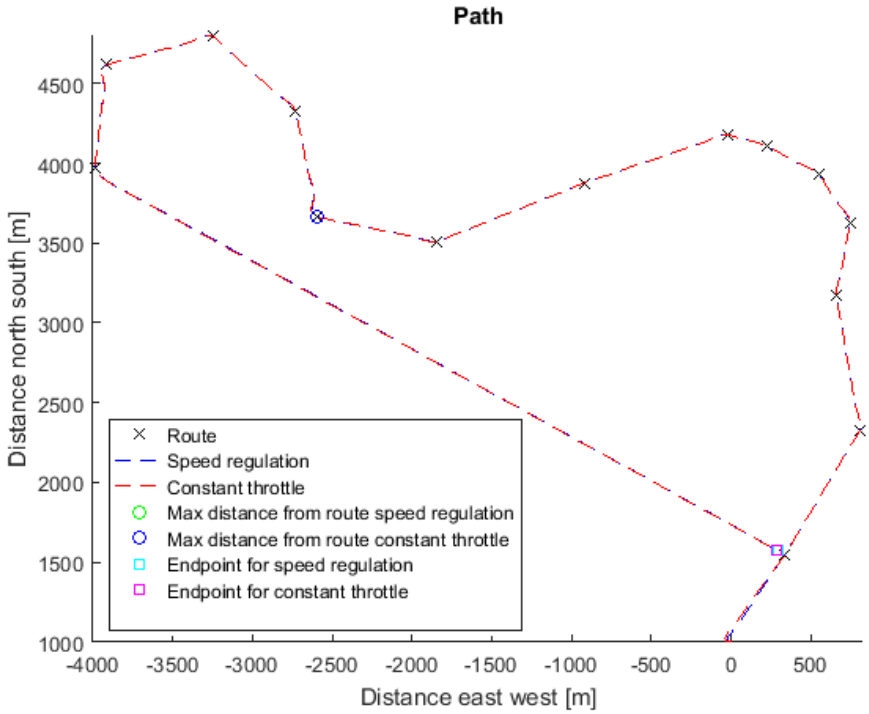

Figure 4.10: Normal route without current. Maximum route deviation with variable throttle is 2.06 [m]. Average route deviation with variable throttle is 1.29 [m]. Maximum route deviation with constant speed is 1.75[m]. Average route deviation with constant speed is 1.06 [m].

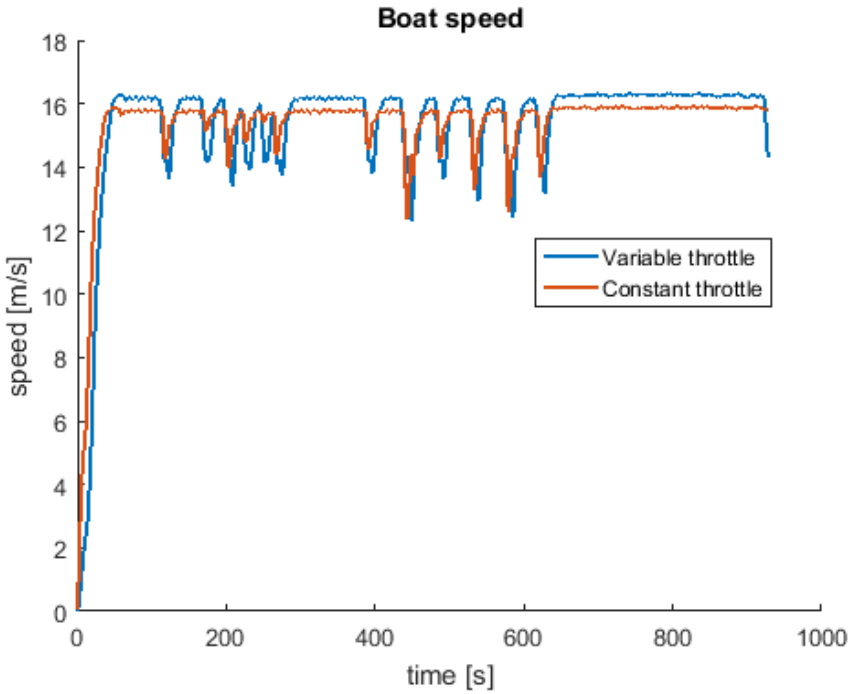

Figure 4.11: Comparison of longitudinal velocity between variable and constant throttle on the normal route without static current.

All presented runs have been made without added noise to the GPS signal, because the difference between the runs with and without the added noise was, as can be seen below in Figure 4.12, insignificant. The noise was ruled out to be considered a problem in the real life testing and no further actions were taken regarding this matter.

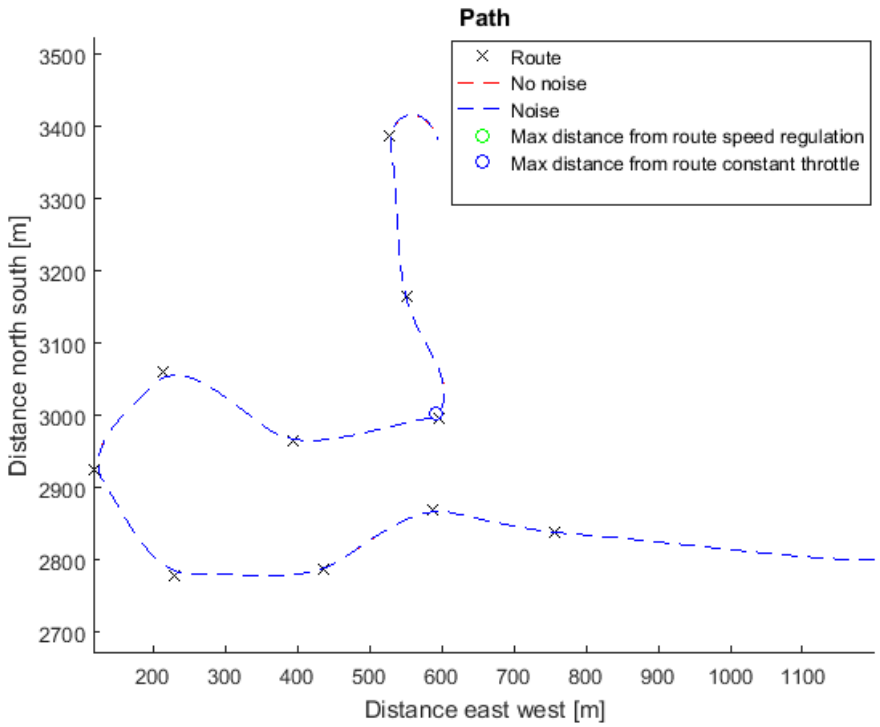

Figure 4.12: The curvy route driven with and without added noise to the GPS signal. Little to no difference is noted.

Maximum route deviation for the run without added noise was $7.64m$ whilst the run with noise was as furthest $7.67m$ away from the route. The average distance differed by only $0.04m$, which is not to be considered vital when the average distance already is in the range of roughly $4m$.

4.1.3 Interpolated Points and Pure Pursuit

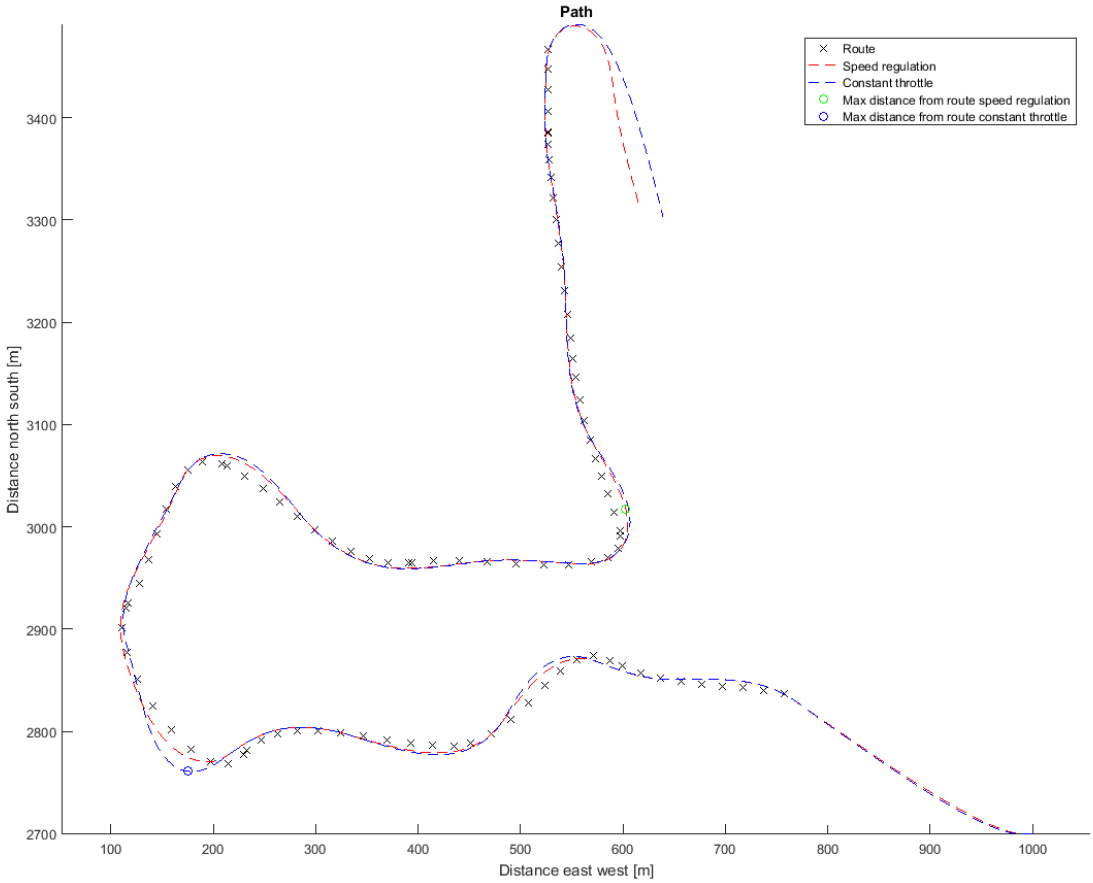

Figure 4.13: Curvy route with interpolated waypoints. Maximum route deviation with variable throttle is 11.65 [m]. Average route deviation with variable throttle is 4.15 [m]. Maximum route deviation with constant speed is 21.55 [m]. Average route deviation with constant speed is 4.95 [m].

In Figure 4.13 the route has been interpolated first in order to smoothen the path. The velocity of the constant throttle is set to match the variable throttle case. The variable throttle algorithm used is the constant lateral acceleration algorithm from Section 2.5.3.

4.2 Testing and Validation

Unlike the simulated resuslts where wind and current were neglected the real life testing will be affected by the forces of nature. Waves, boat size and what type of engine is used will also affect the tests. Despite this it is still possible to draw conclusions and compare the simulations and real life testing.

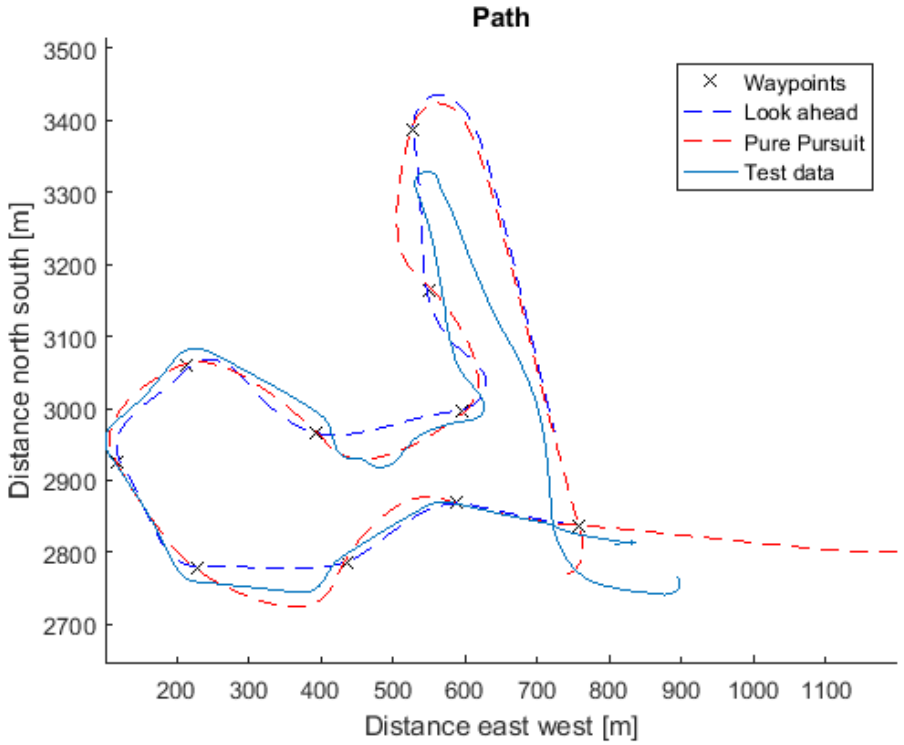

Figure 4.14: Comparison between simulation and test data.

In Figure 4.14 the constant throttle for the *pure pursuit* and *look-ahead* steering algorithms from the simulation environment is compared to a Garmin steering system combined with the constant acceleration velocity algorithm in a real sea test. The reason for the path being shorter in the real test was that it was aborted before the second to last waypoint. By looking on the plots from the empirical test and comparing it to the different simulations it was concluded that the Garmin steering system uses an algorithm similiar to the *pure pursuit algorithm*.

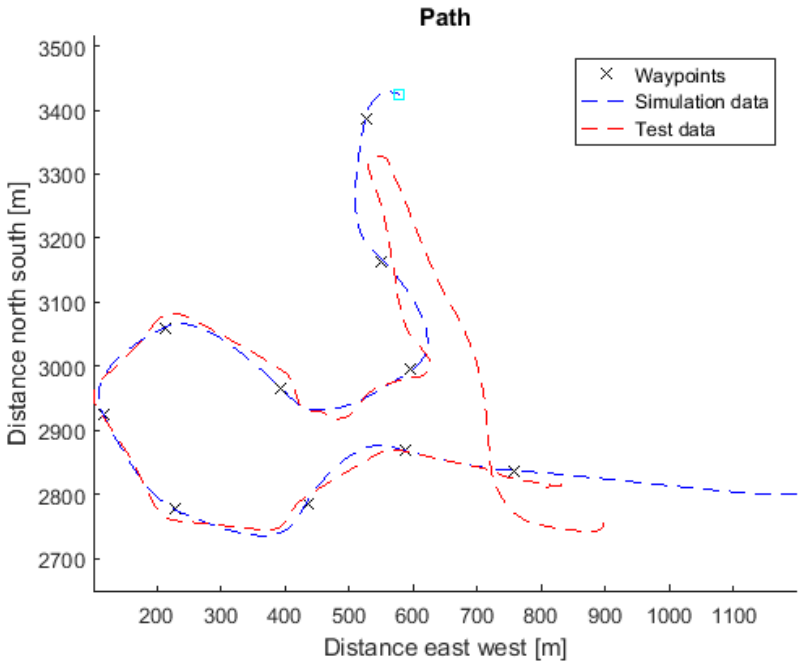

Figure 4.15: Comparison between *pure pursuit* steering algorithm with variable throttle from the constant lateral acceleration algorithm to test data using a Garmin steering system with constant lateral acceleration algorithm.

In Figure 4.15 the two curves do not match perfectly. It is important to keep in mind that the simulation environment do not take into account external factors, such as wind and waves, which were present on the test. Furthermore the steering systems are not identical, even though they seem to be very similar.

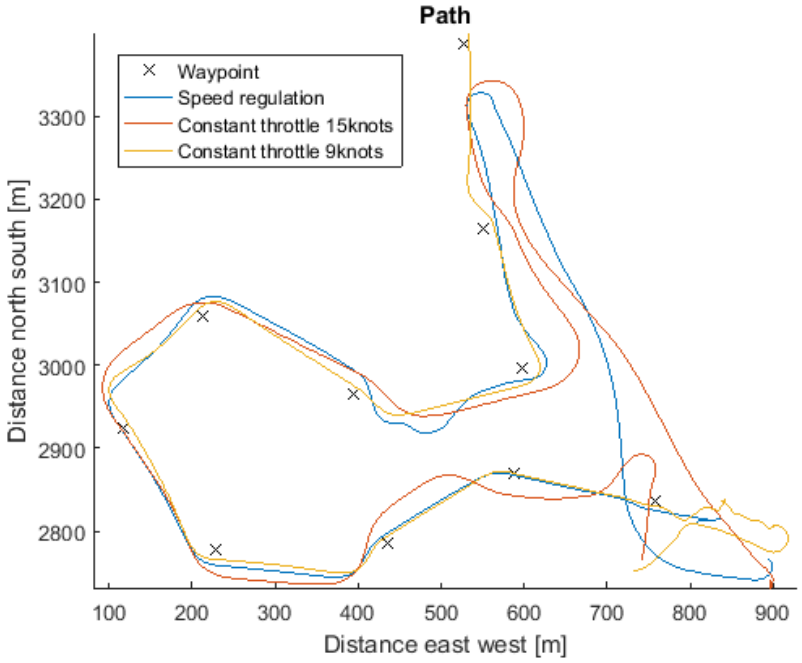

Figure 4.16: Test data from sea trials.

In Figure 4.16 the data from a sea trial using both constant throttle and the regulated throttle is shown. Note that for constant throttle at 15 knots the starting point is different compared to the other two which heavily influences the first waypoints.

5

Discussion

5.1 Steering

From Figure 4.1 it is clear that the *look-ahead algorithm* is far better for path following. The *pure pursuit algorithm* aims straight for the waypoint, meaning that it will come very close to the waypoint but will have a very large overshoot after it since it does not take into account how the route looks after the waypoint. The downside of using the *look-ahead algorithm* is that it will try to cut the corners. The way the algorithm works is that it always looks x meters ahead of the vessel, meaning it will start to turn into the corner earlier and therefore slightly miss it. For this project, however, it has been decided that it is worth cutting the corner to reduce the overshoot significantly. Furthermore, the *look-ahead algorithm* generates a smoother steering pattern. If the *pure pursuit algorithm* slightly misses the waypoint, it will make a heavy turn into waypoint, making the vessel lose bearing towards the path. It is not only undesireable from a path following perspective but also from a comfort perspective.

5.2 Velocity Algorithms

The different velocity algorithms are very similar, the differences between them depend highly on how they are set up. For the two *look-ahead algorithms*, the look-ahead distances will heavily influence how the paths are followed. However, it can be seen that from a path following perspective the comfort algorithm, even though it is based on the same idea, is not working as good as the other two. It is very similar to the constant throttle, but it has a very large overshoot after each waypoint, meaning that it simply does not slow down enough in order for it to turn for the next waypoint. The main reason is that it is purely reactionary nad has no prediction, meaning that the algorithm reacts when it is too late. There are no results of the path following algorithm, the reason is that it is also a purely reactionary algorithm, since the lateral force needs to be taken into account. If the steering algorithm was to be developed in conjunction with the velocity algorithm, the model showed great promise.

5.3 Path Following

Examining the results the numbers for constant throttle and regulated throttle, the results differ only slightly or are in favor of the constant throttle. The phenomena is expected and can be derived from how the steering reference algorithm works. In Figure 4.6, the *look-ahead algorithm* was used. The *look-ahead* will by definition cut corners, for a look-ahead distance greater than 0. If it is assumed that the vessel always has ideal tracking, meaning that turning capabilites are perfect, the resulting path would still have a significant distance to the waypoints.

The reason that constant throttle has "better" results is that with constant throttle, the turn is overshot, meaning that when the *look-ahead algorithm* starts to cut the corner the constant throttle simply overshoots instead.

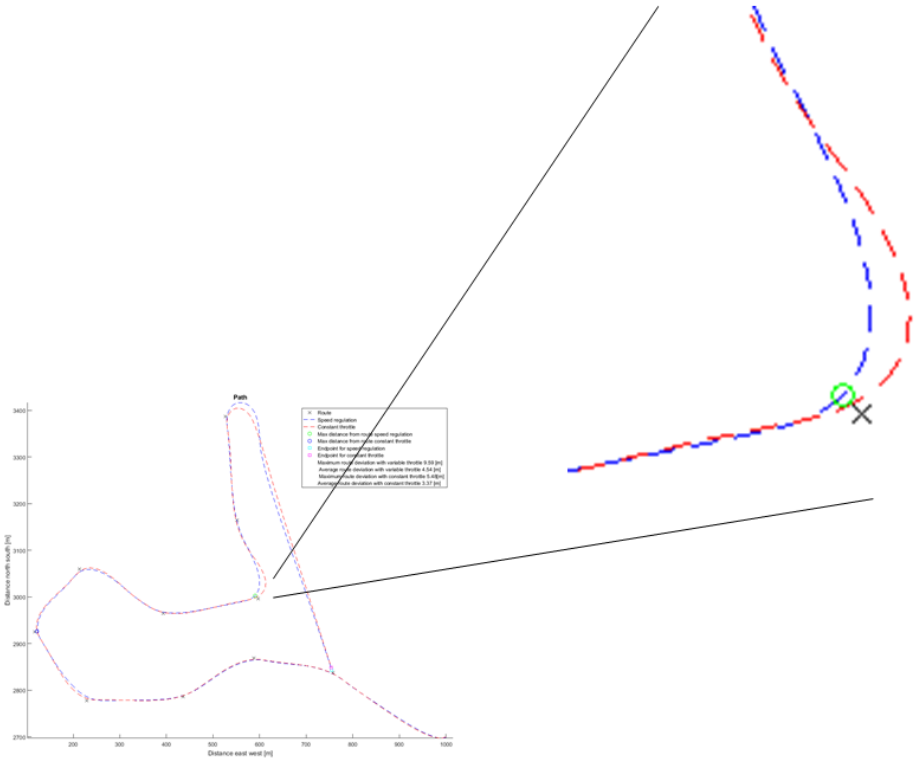

Figure 5.1: Closer look on the curvy path with no current, Figure 4.6, red lines are constant throttle and blue lines are regulated throttle.

In Figure 5.1, it is clearly visible that the red line, representing constant throttle, is closer than the blue line, representing regulated throttle to the waypoint. Keeping in mind, however, that the steering algorithm tried to cut the corner, that is, to turn in *earlier* than the waypoint, the resulting path is not more accurate for the constant throttle. It is very clear that the reason that the constant throttle is closer to the waypoint than the regulated one is that it has too high velocity during turn in and overshoots the corner. Futhermore, the steering regulator is not fast enough

to turn in the vessel, comparing the rudder angle to the steady state rudder angle for the same velocity, it is not large enough to achieve the same radius.

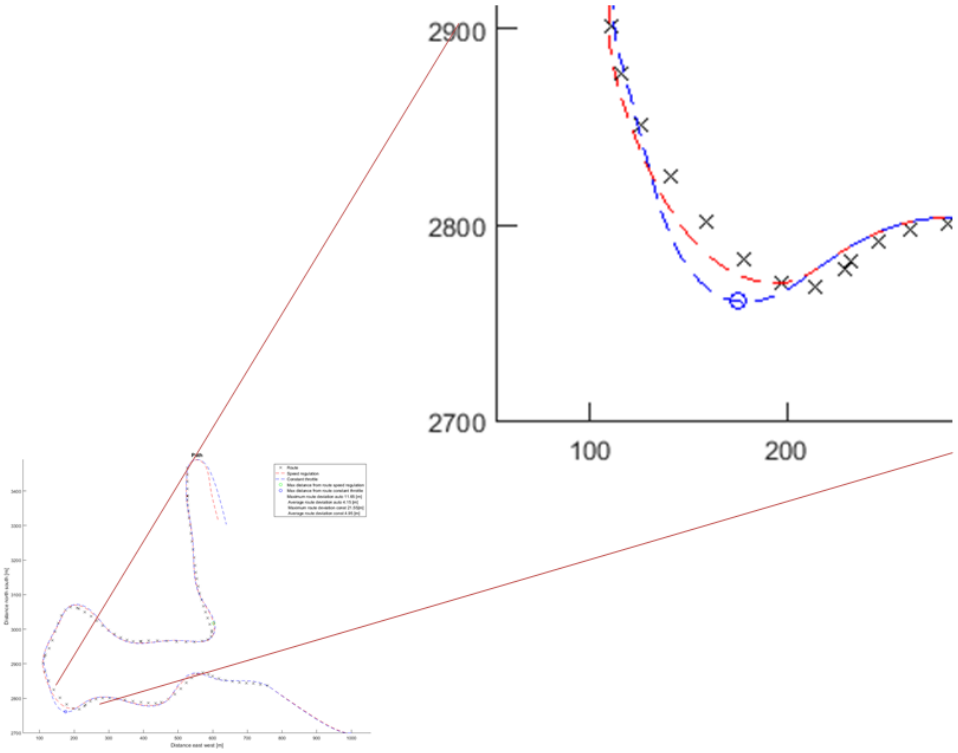

Figure 5.2: Closer look at curvy path for two look-ahead steering. Blue lines are constant throttle and red lines are variable throttle

Now consider Figure 5.2, where both the constant throttle and the regulated throttle overshoots the turn. The reason is mainly that the rudder angle is not sufficiently large. The regulated throttle do, however, help to turn in the vessel, but does not completely solve the issue.

When looking at Figure 4.10. The results are very similar, both the maximum deviation and the average deviation differ with approximately 30cm. The reason for this is that during ideal conditions, that is no external factors, the route can be taken with a low enough constant throttle. That means for a given route, if the throttle is sufficiently low, the need for a variable throttle is not needed. However, as seen above, the need of adjusting the velocity has a very large effect on turns when the velocity is too high.

5.4 Lateral acceleration

As mentioned in Section 2.5.3 the lateral acceleration is set to a constant value for one of the path following algorithms. The value of the constant acceleration is set

according to desired comfort and path following ability. However, it still has to be tuned for each and every boat model individually because different boats behave differently in the water. The most accurate measure of comfort is true turn, which means that desired lateral acceleration is a function of how much the boat rolls while turning. One measurement that is common to categorize a turn is in degrees deviant from 100% true turn [11]. The guidelines are set as in Table 2.2 in Section 2.5.1

Together with Equation 2.29 it can be understood that a boat that is more prone to roll will be more comfortable in high velocity corners than a boat that is rolling less, as long as the boat is not rolling too much. To conclude, there is no constant value on the lateral acceleration that is desirable for all types of boats. Instead, it is proven to be very individual, not only for different boat types but also for what a passenger percieves as comfortable.

5.5 Testing and Validation

During the test of the algorithm, the constant lateral acceleration algorithm reduced the velocity before the actual turn. Even when lowering the look-ahead distance, the algorithm slows down too early. One of the main reasons for this is that the way the Garmin steering algorithm works compared to the *look-ahead* steering algorithm works is that the Garmin steering system turns in very late, which makes it undesirable to look too far into the path, while the *look-ahead* steering algorithm turns in earlier, meaning that the vessel need to slow down earlier. One way of solving this could be to instead of using two points in front of the vessel, use one in front and one behind. Unfortunately, this was not possible to test due to time constraints. However, using a point behind the vessel would mean that the vessel would slow down before the turn and not speed up again until the point behind has passed the waypoint as well. Another possible solution could be to modify the look-ahead function, $l(v)$ in a way such that it takes into account where the vessel is compared to the waypoint and allowing it to become negative.

From Figure 4.16 it is clear that the difference between 9 knots and the regulated speed is negligable, for reasons mentioned above, i.e. that the algorithm slowed down too early and hence did not affect the capability to turn.

Furthermore, the modelled vessel is not the same as the one tested in. The vessels are however, relatively similar, both have IPS systems and are of similar length, but the weights of the vessels differ significantly. The desired methods however, were not intended to be algorithms for one certain vessel, meaning that the algorithms should work independently of the vessel.

6
Conclusion

The growing automatisation of today's society includes the marin industry as well. Autopilots for boats is an old concept. However, the autopilots only control the steering. Existing navigation systems, e.g. plotter systems, can predict and calculate a route. As plotter systems become more advanced, the more detailed routes can become. Current advanced systems can create routes from one point to another, while avoiding obstaces such as islands and shallow water. The current autopilots does not follow a route perfectly and thus could hit obstacles.
The project has tried to enhance autopilots by incorporating speed control as well. The project has also included creating a simulation environment where the autopilot could be tested. A simulation model was created in Matlab/Simulink where the physics were simulated togheter with a graphical representation in Unity. This ment that real time testing could be done. The simulation environment was successful, it was accurate enough to simulate the behaviour of the vessel to create an algoritm for speed control. The aim to enhance the autopilots of today instead creating a complete new one ment that commercial autopilot needed to be used to control the steering while an additional algoritm, created in this project, controlled the throttle of the vessel. Due to the fact that it was not possible to use a commercial steering algoritm in the simulation environment, a steering algoritm was developed in the project as well. In the simulation enivornment where both the steering and the speed algoritm could be tuned, it was possible to enhance the path following ability of vessels. In real world testing however, the commercial steering algoritms were not developed to do multipoint routes but only going from point A to point B. Furthermore, each autopilot is different, it will not be possible to create an algoritm which will work as an addition to the current auto pilots which enhance the path following ability to a satisfactory level.
Further work should try to instead of enhancing the current autopilot system, try to create a new type of autopilot which includes both steering and throttle control. It is clear that such an algoritm would enhance the path following ability.

Bibliography

[1] Frisk, D. (2016) A Chalmers University of Technology Master's thesis template for LaTeX. Unpublished.

[2] D. Higdon (2010, May) Avionics News [Online]. Available: https://www.aea.net/AvionicsNews/ANArchives/MainFeatureMay10_LetGeorgeDoIt.pdf, 2017-01-18.

[3] TMQ International. How does an autopilot work [Online]. Available: http://tmqeurope.com/information/how-does-an-autopilot-work/, 2017-01-18".

[4] GetMyBoat. Autopilot Systems on Boats [Online]. Available: https://www.getmyboat.com/resources/tips-for-owners/471/autopilot-systems-on-boats, 2017-01-18".

[5] T.M Jensen, "Waypoint-Following Guidance Based on Feasibility Algorithms", Norwegian University of Science and Technology, Trondheim, 2011.

[6] Press, W. H.; Flannery, B. P.; Teukolsky, S. A.; and Vetterling, W. T. Numerical Recipes in FORTRAN: The Art of Scientific Computing, 2nd ed. Cambridge, England: Cambridge University Press, p. 710, 1992.

[7] K. A. N Jørgensen,State Estimation with Wave Filtering for an Unmanned Surface Vehicle, Norwegian University of Science and Technology, Trondheim, 2011.

[8] U. Holmberg PID control - Simple tuning methods http://www.hh.se/download/18.7502f6fd13ccb4fa8cb2f6/1360676928481/PID.pdf, 2017-06-05"

[9] B. Lennartson, Reglerteknikens grunder, Studentlitteratur, 2002.

[10] http://www.openstreetmap.org/copyright, 2017-06-05.

[11] T. Magnusson. Volvo Penta. Meeting: A boats behaviour and important parameters (2017, February 27)

[12] H.I.H. Saravanamuttoo, G.F.C. Rogers, H. Cohen, Paul Straznicky, "Gas Turbine Theory", 6th ed, England, Pearson Ed Limited, 2009