Development of autonomous navigation systems for maritime applications

Mathias Storbacka

Master’s Thesis in Process and Systems Engineering

Instructors: Senior Lecturer Jari Böling, Project Manager Kai Jämsä

Faculty of Science and Engineering

Åbo Akademi University

September 2021

Abstract

The interest in autonomous technology for transportation has grown significantly in the last decade and is slowly reaching the maritime industry as well. There are numerous potential benefits in using self-driving vehicles for maritime applications, both regarding performance and economy, as well as safety. Autonomous systems for maritime vehicles are still in an early phase of development and commercially available solutions are yet to come. This thesis discusses autonomous navigation systems for maritime applications, with focus on waypoint tracking and autopilot design.

The Department of Computer Science at Åbo Akademi University has started a project for developing an autonomous boat, ÅBOAT, which the autopilot system described in this thesis is specifically designed for. Specific emphasis is placed on researching and developing smooth path generation for waypoint tracking autopilot systems and three different smoothing interpolation methods are compared. Other elementary functions of autopilot systems and their waypoint tracking algorithms are also discussed and developed.

Keywords: Autonomous shipping, waypoint tracking, autopilot

Table of contents

List of symbols

$\boldsymbol{C}$	Matrix for Coriolis and centrifugal forces
$\mathrm{E}(x)$	Gaussian expectation operator
$\boldsymbol{F}$	State transition matrix of a dynamic system
$\boldsymbol{G}$	Dynamic system model parameters
$\boldsymbol{H}$	Sensor model parameters
$\bar{\boldsymbol{K}}$	Kalman gain matrix
$\boldsymbol{M}$	Inertial mass matrix
$\boldsymbol{O}$	Covariance matrix of measurement uncertainty
$\boldsymbol{P}$	Covariance matrix of state estimation uncertainty
$\boldsymbol{Q}$	Covariance of process noise in system state dynamics
$\boldsymbol{R}$	Rotational matrix
$\boldsymbol{v}$	Random sensor noise
$\boldsymbol{v}_s$	Linear velocity of ship given in the body-fixed frame
v_x	Surge velocity
v_y	Sway velocity
v_r	Yaw rate
$\boldsymbol{w}$	Random dynamic disturbances
$\hat{\boldsymbol{x}}$	State vector
$\boldsymbol{z}$	Sensor outputs
$\boldsymbol{\tau}$	Force vector
$\boldsymbol{\eta}$	Position and rotation in the world reference frame

Γ	Thrust configuration matrix
φ	Latitude coordinate in radians
λ	Longitude coordinate in radians
θ	Compass bearing

Abbreviations

IMU	Inertial measurement unit
3DOF	Three degrees of freedom
LIDAR	Light detection and ranging
PID	Proportional-integral-derivative controller
LQR	Linear-quadratic regulator

1 Introduction

Autonomous technology, robots and automated processes have been a central part of the society for a long time. In the most recent decade, the interest in autonomous technology has reached the transport industry as well, and the research and development of self-driving vehicles have grown significantly. This can be observed particularly in the automotive industry, where some autonomous vehicles already are available on the public market and automatic features such as parking assistants have become ordinary equipment in cars. Significant research and development are also conducted in the maritime industry. The target group is, nevertheless, mainly the shipping industry, and autonomous maritime technology has therefore gained less publicity compared to the automotive industry, where the products are available on the public market. There are, however, many possible areas of use for autonomous maritime vehicles where the benefits would be substantial.

The required sensor technology, navigation- and network infrastructure and other related hardware needed for the development of autonomous shipping are already available. However, combining these elements in a reliable and cost-effective way is challenging and extensive testing and simulation is required to create ship manoeuvring algorithms that make suitable decisions under all circumstances. Another challenge in introducing autonomous shipping are the legal implications. The maritime law is currently best suited for manned vessels, and additional rules and regulations would have to be adopted to cover the operation of autonomous vessels.

The focus of this thesis is on studying autonomous maritime navigation systems and developing software for the ÅBOAT-project at the Department of Computer Science at Åbo Akademi University. Although autonomous maritime navigation systems will be discussed as a whole, the primary goal is to develop a waypoint tracking autopilot system. Adapting suitable paths and trajectories to waypoints and providing heading- and velocity setpoints for manoeuvring the vessel is the main function of the software that will be developed. The development of the system for steering and thrust control and the positioning system for the ÅBOAT-platform is also discussed in further detail, as the waypoint tracking autopilot is dependent on their design.

2 Autonomous maritime navigation

Autonomous maritime vehicles utilize many of the technologies already implemented in self-driving cars. Situational awareness and collision avoidance can be achieved with cameras, radars, LIDAR and AIS, combined with IMU, compass and GNSS that measure the vessel’s own movement and position. An autonomous vessel is also dependent on communication technologies. Mobile networks, satellite, and radio connections can be used for reporting a vessel’s status to other vessels or control centres on shore.

The implementation of autonomous maritime navigation has several benefits over traditional systems where a crew is required for operation. Starting from a ship layout perspective, autonomous or remote control of a ship eliminates the need for a bridge or cockpit, social spaces as well as water and sewage systems on a ship. The removal of the above-mentioned equipment is beneficial on practically any platform, as transport space thus can be increased. Unmanned ships are especially beneficial in cargo transport, where the cargo space can be greatly increased. Safety is, at least in theory, another advantage of unmanned ships, as there is no ship crew working under dangerous weather conditions. The possibility of human error in ship operation is eliminated as well. The harsh weather conditions, long working periods, and isolation that ship crew often experience are factors that make working on ships less attractive and therefore complicate the recruitment of personnel to the maritime industry, and this problem is solved by autonomous technology as well.

However, as previously mentioned, there are some challenges in developing and utilizing autonomous vessels. Although the safety of workers is enhanced, the safety of the cargo and vessel itself also needs to be guaranteed. Communicating with vessels and monitoring their progress can be difficult as network bandwidth is limited offshore. This sets the needs for a robust autonomous system, which performs well under all circumstances. Offshore service and maintenance are also limited, or at least delayed, without onboard crew, which further sets high demands on the system reliability. Cybersecurity and piracy are challenges, which need to be tackled as well.

As mentioned in the previous chapter, the maritime laws need to be revised to fully cover the operation of autonomous vessels. The International Maritime Organization, IMO, sets the standards for international shipping and provides several conventions for achieving safety in maritime traffic. The rules for safe navigation on sea are clearly defined in the different conventions by IMO and can be implemented in autonomous navigation algorithms, but the use of terms such as “master”, “crew” and “responsible person” make the rules rather ambiguous when applied to autonomous vessels. The current regulations for mandatory equipment regarding safety are only applicable for manned vessels and would therefore need revision as well. IMO is, however, currently revising their current conventions to suit autonomous shipping, and the updates are expected to be ready in the near future.

In other words, there would be numerous benefits in implementing autonomous systems in maritime vehicles but the transition to autonomous technologies is technically challenging and requires several revisions in the structure of maritime traffic. The required technology for building autonomous vessels is already available and the main challenge is to create systems that are reliable enough to withstand harsh weather conditions and operate safely according to good maritime practice under all circumstances.

2.1 Autonomous ferries

Ferries are well suited for the use of autonomous technologies due to their monotonic operating routines. Car and commuter ferries are especially suitable for autonomous technology, as they typically only offer their passengers transport service where human interaction is unnecessary. Autonomous technology could provide benefits with increasing both transport capacity and operating flexibility. A project aiming at this area of use is the AAWA project conducted by Rolls Royce and Finferries. The AAWA project uses a car ferry operating in the Turku archipelago as a platform for developing an autonomous system (Business Finland, 2018).

Ferries provide a good starting point for development of autonomous vessels as the routes are short and consistent. Ferries usually operate at locations close to the shore, which enables relatively stable connections between the vessels and shore control centres via mobile networks. These operating locations also provide possibilities for data collection used for developing object detection which is a vital part of situation awareness on an autonomous platform.

The primary use of the vessel in the ÅBOAT-project is to act as a platform for developing an autonomous system for maritime vehicles for use in urban areas or for short routes in built-up areas. Car- and commuter ferries are good examples of vessels operating in these areas. These areas provide a good and demanding testing environment as there usually is other traffic present and the fairways are narrow and require precision, while the weather conditions usually are adequate, with low wind speeds and calm waters. There are plans on utilizing the ÅBOAT autonomous system on other vessels as well, and ferries operating in urban environments are potential platforms.

3 ÅBOAT-project

The control and navigation systems in this thesis project are specifically designed for a boat platform developed by the Department of Computer Science at Åbo Akademi University as part of the MAST! -project. The boat mainly serves as a piloting platform for developing the autonomous navigation system and a goal for the system is to be configurable to suit other vessels as well. The platform is ought to be operated both fully autonomously, and remotely from a remote operation centre on shore. Compatibility and portability are achieved by using a modular hardware design where the different components consist of publicly available devices with standard connections.

3.1 Platform design and hardware

The platform is developed using an inflatable fishing boat as base. The base of the platform has a length of 3750 mm and a weight of 34 kg, with a maximum load capacity of 295 kg, which is enough for carrying the needed hardware and equipment. Two electric trolling motors are used for propulsion and steering, and these units are mounted in each end of the boat. The propulsion units are of azimuth-type and can be rotated from -180 to 180 degrees, which increases manoeuvrability and also allows for implementation of a dynamic positioning system at a later stage. Equipping the ÅBOAT-project platform with azimuth thrusters is beneficial as they also are commonly used on larger modern vessels, resulting in better intercompatibility of the control system. The steering units are integrated in the chassis of the motors and both motor speed and rotation are controlled via a NMEA2000 bus. NMEA2000 is an industry standard for communication between components in maritime applications. An 80 Ah 12 Volt lead battery is powering the system, which delivers a runtime of approximately 2 hours at cruise speed.

The system utilizes various types of sensors to enable precise monitoring of the platform status. The sensor data is used for achieving situational awareness for the autonomous system and for enabling feedforward control of the propulsion units. The data is also forwarded to the remote operation centre which allows for precise monitoring of the ship’s status on shore. An Nvidia Jetson AGX Xavier AI computer is used for processing data required for object detection and achieving situational awareness. Four cameras connected to a Nvidia Jetson AGX Xavier processing unit are used to achieve a 360-degree view around the platform, and an Ouster OS-1 LIDAR-unit is used for measuring distances to objects within 120 metres. The Nvidia unit further uses positioning data for acquiring the position of the vessel.

A Raspberry Pi single-board computer is used for CAN and NMEA2000 communication with several different devices. The Pi also hosts several microservices, including the waypoint tracking autopilot and the steering and thrust control system. The communication is executed using a CAN-expansion board on the Raspberry Pi which is linked to a network of NMEA2000 specification cables. The two trolling motors are connected to the Pi CAN-bus and controlled via NMEA2000 messages. Additional custom fabricated motor turn angle sensors are connected to the NMEA2000 network as well, enabling more precise steering. A Pololu MinIMU-9 v5 IMU featuring a gyro, accelerometer and magnetometer is connected to the Raspberry Pi via I^2C and provides compass and inertial data for the system. The IMU is also used in combination with GNSS data for more accurate and responsive positioning of the vessel. Additional motor sensors are used to obtain the position and heading of the platform. The CAN-bus allows for expansion of the system, and a wind sensor is planned to be installed, improving the system with feedforward steering and thrust control.

Connection to the shore control centre is handled with a Teltonika RUTX-11 LTE-router connected to the shore via an IPsec tunnel. The LTE-router also features GNSS which is used to obtain positioning data, COG, and ground speed. Onboard communication between the Nvidia Jetson AGX Xavier, Raspberry Pi, Ouster LIDAR-system and LTE-router is handled via ethernet connections. A detailed picture of the system layout can be seen below.

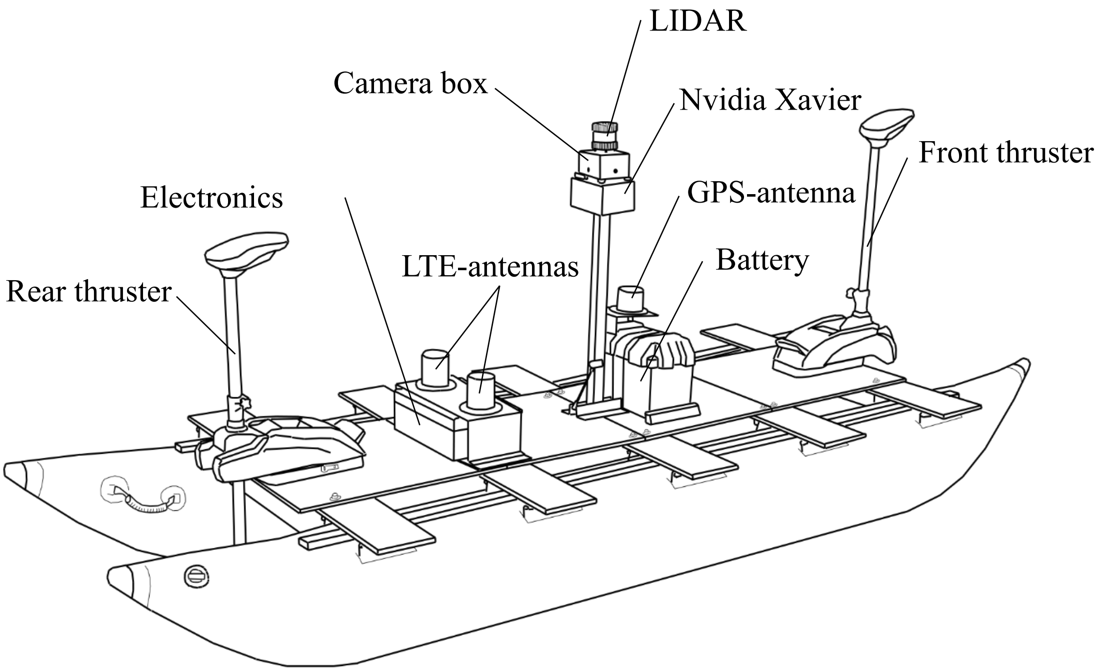


Figure 1. Hardware layout and configuration

3.2 System description

The autonomous navigation system on the ÅBOAT-platform is constructed out of several microservices running both on the platform hardware and on the shore control unit. The system is developed using the OpenDLV framework, which is a microservice-based software ecosystem for self-driving vehicles (OpenDLV, 2020). The microservices have varying functionalities, ranging from providing sensor data, to performing more comprehensive computations used for manoeuvring the ship. The focus of this thesis is on mainly on the high level microservices and functionalities that form the autonomous navigation system, primarily concentrating on the autopilot system.

The high-level microservices of the autonomous navigation system on the ÅBOAT are, the route planner, waypoint tracking autopilot, motor controller, motor driver, situational awareness module and the collision avoidance system. The autonomous navigation system additionally relies on several other microservices which supply sensor data and manage the communication between microservices. These other microservices will however, not be discussed in further depth, except for a sensor fusion microservice which includes a Kalman filter for determining the current state of the vessel. Kalman filtering is essential for obtaining reliable and real-time position data and will be discussed more comprehensively in a later chapter. A block diagram presenting the architecture of the high-level components in the autonomous navigation system is presented below.

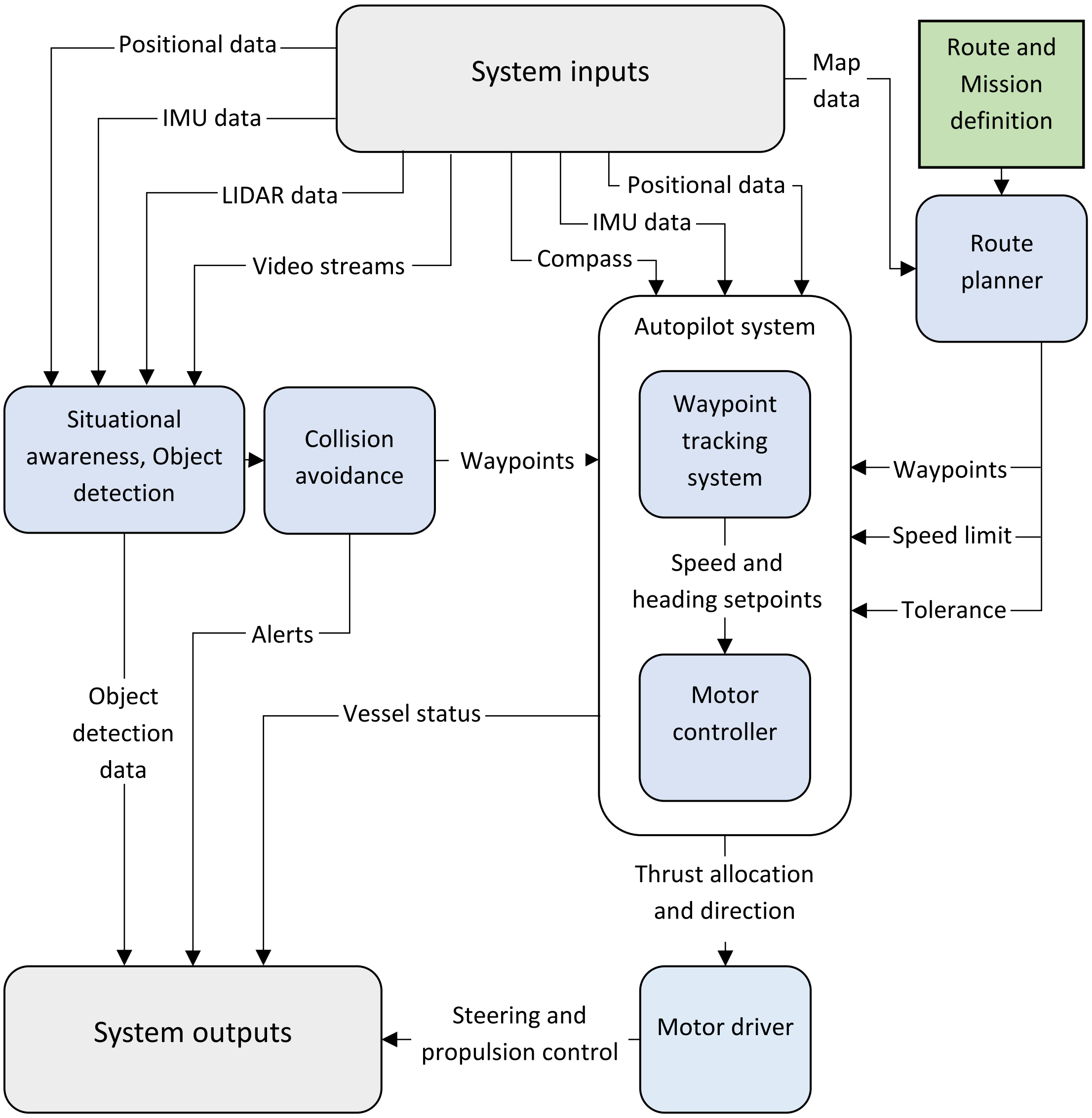


Figure 2. System architecture diagram

4 Autonomous navigation system

The autonomous navigation system on the ÅBOAT-platform consists of four main components: a route planner, waypoint tracking autopilot and a control system for thruster allocation, as well as a collision avoidance module. The components together form a system that can manoeuvre a vessel autonomously from one location to another. The route planner defines a route in the form of waypoints, and the waypoint tracking autopilot provides manoeuvring commands for the motor control system. The motor control system converts heading- and speed setpoints to actual steering and throttle inputs for the motors using PID control technology. The waypoint tracking autopilot reads position data from GNSS and ensures that the vessel moves along the planned route. The collision avoidance unit utilizes the cameras and sensors on the platform and alters the route to avoid colliding into non-static obstacles that are unknown to the route planner.

Although the focus of this thesis will be on the design of the waypoint tracking autopilot, the design, features and requirements of the route planner, motor control system and collision avoidance unit will be discussed briefly as well. Although ship modelling theory is mainly linked to the design of steering and thrust control systems, elementary understanding of it is useful whenever designing systems for use in maritime vehicles. The essentials of ship modelling in 3DOF (three degrees of freedom) will therefore also be studied and summarized in the steering and thrust control section.

4.1 Path planning

The path planning is done on a major scale by the route planner, which generates a series of waypoints for the entire journey. The waypoints consist of coordinates given in a specific order that form a route when linked together. The waypoint coordinates are determined through the use of map data containing information on fairways and water depth and local speed limits. The path planner also sets manoeuvring tolerance for each waypoint depending on the width of the fairway or overall room for error at the waypoint locations. As the route planner module generally provides waypoints in intervals of several hundred meters, further route details are required for precise manoeuvring in between waypoints. These additional route details are generated by the waypoint tracking autopilot using interpolation, which results in additional intermediate waypoints.

An automatic route planner is yet to be designed for the ÅBOAT-platform. There are, however, several nautical chart providers who already have this option available in their services and these systems can be useful when designing a system for the ÅBOAT-platform. Openseamap is an opensource project which provides free nautical charts for navigation on sea and is used in the ÅBOAT-project due to its good integration possibilities. A semi-automatic route planner that is included in Openseamap, in combination with an XML to JSON converter, has been used for planning the routes used for testing and development of the autopilot system. The need for an automatic route planner is currently uncertain, as the ÅBOAT-platform most probably will traffic at specific predefined routes. An automatic route planner is, however, a useful feature and will most likely be developed for the ÅBOAT-platform soon.

4.2 Waypoint tracking autopilot

A central part of the autonomous navigation system is the waypoint tracking autopilot, which transforms waypoints to path trajectories for the vessel and provides manoeuvring commands for the steering- and thruster control system. These trajectories are obtained by interpolating between the waypoint coordinates, which results in sets of mathematical functions that describe the path. Two common methods used for waypoint interpolation are cubic spline and cubic hermite interpolation (Fossen, 2002). The desired heading is then calculated from the trajectory function and passed on to the motor controller system which converts the messages into manoeuvring actions. The waypoint tracking autopilot also supplies the motor control system with the desired velocity, which origins from map data and local speed limits. The speed command can be also be altered if a lower speed is required for keeping the ship on the desired trajectory, e.g., in a tight turn. The complete design of the waypoint tracking autopilot system for the ÅBOAT-platform is further explained in detail in Chapter 5 of this thesis.

4.3 Steering and thrust control

The heading and velocity setpoints given by the tracking autopilot are processed by the steering and thrust control system which uses suitable control technologies to achieve these setpoints. The development of the control system used in the ÅBOAT-project was conducted as bachelor's thesis work (Fröjdö, 2021) and the design choices were obtained through simulating the behaviour of the boat platform with Simulink, which is a commonly used tool for modelling, simulating, and analysing dynamic systems. A Simulink model for the platform was constructed utilizing three degrees of freedom and control systems were added for controlling the ship propulsion units. Two different controller types PID and LQR were tested and tweaked to perform well with the boat by adjusting different parameters of the controllers in the modelling software and the best performing control system were then used on the ÅBOAT-platform. The actual ship steering and thrust controller module is implemented as a separate microservice and transfers steering input and throttle data to a motor driver which performs the actual motor control via the NMEA2000 bus.

4.3.1 Ship modelling

As the development of the control systems for the ÅBOAT-platform is conducted through simulation, a model representing the behaviour of the platform is needed in order to simulate the performance of the different control systems. The ÅBOAT-platform that the control system is implemented on is light and small in comparison to the vessels that are commonly studied in literature, which complicates the process of finding data for the vessel's characteristics. Initial modelling and simulation therefore require estimation of ship properties. The low mass and low cruise speed of the ÅBOAT-platform also implies that the impact of the Coriolis and centrifugal components of the ship dynamics modelling are negligibly small, and these components are therefore excluded from the model. The impact of wave induced forces is difficult to evaluate onboard the ship and are thus excluded as well. Inclusion of these wave induced forces in the feedforward control would only result in little to no benefit in control response. All other factors, however, have been considered when modelling the vessel's dynamics, regardless of the magnitude of their impact in this application. The ship modelling can be divided into two parts, kinematics, and kinetics. Kinematics are used to describe the movement of rigid bodies without considering the forces causing the motion (Beggs, 1983), while kinetics explains the impact of forces acting upon rigid bodies.

4.3.2 Ship kinematics

The ship is assumed to be a rigid body object. The movement of a rigid body has six degrees of freedom; translation and rotation on three coordinate-axes. Translations along the axes are referred to as surge, sway and heave and the rotations are called roll, pitch, and yaw (Åström, Källström, 1976).

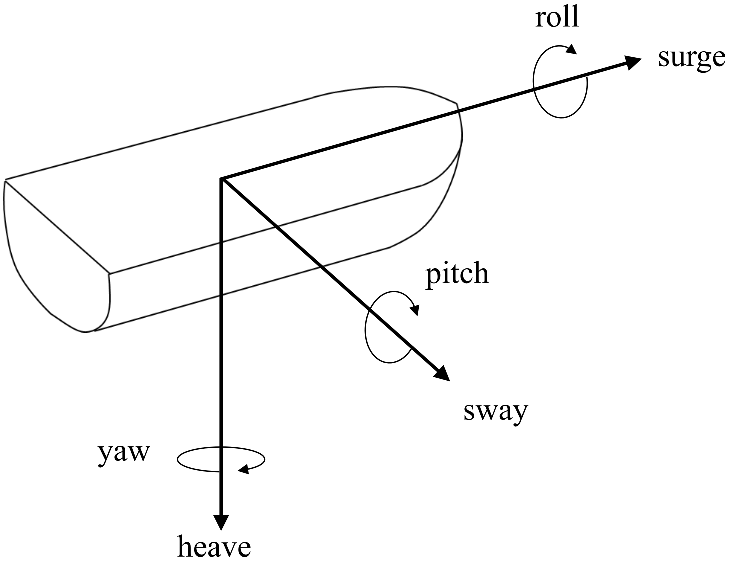


Figure 3. The 6 degrees of freedom of a rigid body

In this thesis however, it is assumed that the ship's movement is restricted only to the horizontal plane. This assumption results in a simpler kinematics model utilizing movement in only three degrees of freedom; surge, sway, and yaw. Two reference frames are used to describe the movement: the world reference frame, $\boldsymbol{o}_n$, with the coordinate axes north and east, and the body-fixed frame, $\boldsymbol{o}_b$, which has its origin in the ship's centre of gravity. A rotational matrix $\boldsymbol{R}$ is used for translation of the movement from the body fixed frame to the world reference frame. The rotation matrix will be described further in the next chapter.

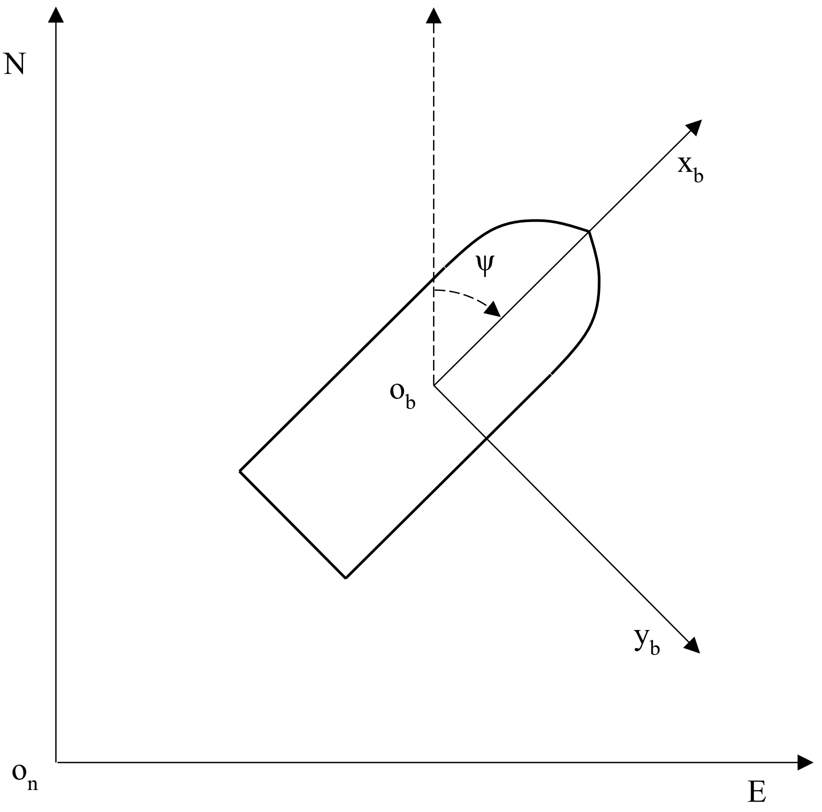


Figure 4. Simplified model with three degrees of freedom

4.3.3 Ship kinematics

A widely used model for describing ship dynamics, is a model presented by Fossen (Fossen, 2011). The following model utilizes three degrees of freedom and is formulated as follows:

$$\dot{\boldsymbol{\eta}}_s(t) = \boldsymbol{R}(\boldsymbol{\eta}_s(t))\boldsymbol{v}_s(t)$$

$$\boldsymbol{M}_s\dot{\boldsymbol{v}}_s(t) + \boldsymbol{C}_s\big(\boldsymbol{v}_s(t)\big)\boldsymbol{v}_s(t) = \boldsymbol{\tau}_s(t) + \boldsymbol{\tau}_{drag}(\boldsymbol{v}_s(t), \boldsymbol{\eta}_s(t)),$$

where $\boldsymbol{\eta}_s(t) = [x(t), y(t), r(t)]^T$ is the position and rotation of the ship in the world reference frame, $\boldsymbol{M}_s$ is the inertial mass matrix, $\boldsymbol{C}_s(.)$ is the Coriolis and centrifugal matrix, $\boldsymbol{v}_s(t) = \left[v_x(t), v_y(t), v_r(t)\right]^T$ is the ship's surge, sway and yaw speed in the body reference frame, and $\boldsymbol{\tau}_{drag}(.)$ is a function for representing drag forces. As mentioned in the previous chapter, the Coriolis and centrifugal forces are negligible for small and slow vessels, and $\boldsymbol{C}_s(.)$ can therefore be excluded from the model used in the ÅBOAT-project. $\boldsymbol{R}$ is the rotational matrix, which transforms the movement in the body-fixed frame to inertial velocities in the world reference frame.

$$\boldsymbol{R} = \begin{bmatrix} \cos(r) & -\sin(r) & 0 \\ \sin(r) & \cos(r) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

τ_s is the vector of forces applied to the ship's centre of gravity by its propulsion units.

$$\boldsymbol{\tau_s} = \begin{bmatrix} \tau_x(t) \\ \tau_y(t) \\ \tau_r(t) \end{bmatrix}$$

The vector τ_s can contain different elements depending on the number and type of actuators on the ship. This thesis will mainly be focusing on ships equipped with azimuth thrusters, which results in the following construction of the matrix τ_s

$$\boldsymbol{\tau_s} = \Gamma(\propto_1, \cdots \alpha_m) \begin{bmatrix} g_1 \\ \vdots \\ g_m \end{bmatrix}$$

where Γ is the thrust configuration matrix, which is dependent of the steering angles $\alpha_1, \cdots, \alpha_m$ of each propulsion unit.

$$\Gamma = [\boldsymbol{\gamma}_1(\alpha_1), \cdots, \boldsymbol{\gamma}_m(\alpha_m)]$$

$$\boldsymbol{\gamma}_1(\alpha_1) = \begin{bmatrix} \cos(\alpha_1) \\ \sin(\alpha_1) \\ l_x \end{bmatrix}$$

A Simulink model of the ÅBOAT-platform and its operating environment can be constructed utilizing the models above. The Simulink model serves as a development platform for the control systems on the ÅBOAT-platform.

4.3.4 Controller design

Controllers of the types PID and LQR, were tested for the ÅBOAT-platform through simulation. Three separate PID controllers are needed for independently controlling surge, sway, and yaw. The LQR-control system controls all three variables simultaneously. An LQR system controls the system parameters by solving an optimization problem that is based on a linear system model, which describes the relationship between system inputs and outputs (Häggblom, Böling, 2013). A linear quadratic controller is capable of multivariable control and cross-coupling between system parameters. Cross-coupling allows the control system to consider the impact of control actions on multiple variables. However, according to Fossen, cross-coupling unfortunately reduces system robustness when the vessel configuration is changed (Fossen, 2002).

The simulations performed with the ÅBOAT Simulink model, proved that both PID and LQR were capable of controlling the system. The setup of the LQR system was, however, more difficult to tune and performed worse with varying system mass. As LQR control requires an accurate system model which is system specific, it is less suitable for cross platform use. Due to the tuning difficulties, reduced cross-compatibility, and lack of robustness of LQR, the final choice of control system for implementation on the ÅBOAT-platform was a PID control system. PID controllers were proven to be stable and easy to setup and offer satisfactory performance for the intended application. The use of PID controllers also increases system compatibility, as there are fewer setup parameters. More detailed information on the design of the ÅBOAT steering and thrust control system can be found in the thesis by Fröjdö (Fröjdö, 2021). A block diagram of the control system is pictured below.

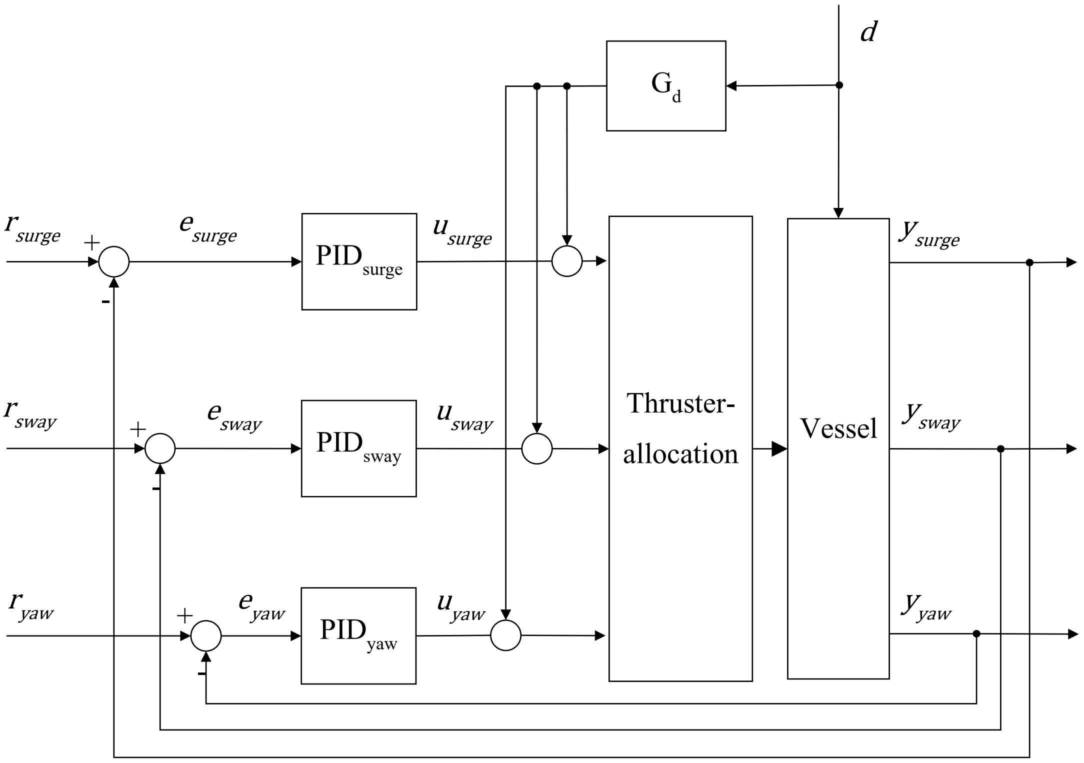


Figure 5. Block diagram of steering and thrust control system

The block diagram pictures the three separate PID-controllers that employ both feedback and feedforward control. The surge and sway controllers form the speed control, and the yaw controller controls the heading of the vessel. The control system utilizes feedforward control to compensate for disturbances that affect the system, such as e.g., wind disturbance. The disturbance signal is marked with a d in the block diagram.

4.4 Collision avoidance

Manoeuvring a vessel on a route where other marine traffic is present, requires additional precautions and general rules must be followed for achieving safe navigation. There are several types of situations where a vessel must deviate from its planned path or slow down to avoid the occurrence of a collision. These manoeuvres should be done in a predictable manner according to general regulations, minimizing the risk of collision. The Convention on the International Regulations for Preventing Collisions at Sea, COLREGs, are a set of regulations that apply to all vessels navigating on international waters as well as in areas connected to international waters (International Maritime Organization, 1972). COLREGs include manoeuvring rules for collision avoidance, as well as information on technical requirements of vessels and requirements on ship crew.

A collision avoidance system on an autonomous vessel is ideally designed to avoid dangerous situations by manoeuvring according to the regulations in COLREGs. The system utilizes regulations in COLREGs regarding overtaking, head-on navigation, crossing paths, giving way for other vessels, and stand-on situations. All these navigation situations require algorithms that output predictable and safe manoeuvring decisions.

The collision avoidance system requires real-time information of the vessels state and its surroundings for determining the risk of collision and possibly intervene in the process of navigating the vessel. Data regarding the vessel's state consists of positioning data obtained through satellite navigation systems combined with IMU, speedometer and compass, as well as data on the planned trajectory of the vessel. The data of the surroundings is obtained with 360-degree cameras, LIDAR and AIS.

Suitable software that utilizes sensor fusion is then used to identify objects in the surrounding and determine their locations and movement. Static map data, containing information on water depth and fairway location, is also needed to ensure that the vessel navigates in safe areas. The acquired data is sufficient for achieving situational awareness and calculating possible risks, which together act as a base for making eventual diverging manoeuvring decision to avoid collisions.

Collision avoidance on the ÅBOAT-platform is designed using the above-mentioned systems and regulations. The system is running onboard the platform and communicates with both the route planner and waypoint tracking autopilot. Route data is received from the route planner and exact trajectories are received from the tracking autopilot. New sets of waypoints are generated whenever the collision avoidance unit recognizes a potential risk, and the updated set of waypoints are then sent to the waypoint tracking autopilot.

5 Design of the waypoint tracking autopilot

This chapter covers the development of the waypoint tracking autopilot designed for the ÅBOAT-platform. The waypoint tracking autopilot on the ÅBOAT-platform is constructed as a separate microservice which runs independently on the Raspberry Pi single-board computer on the platform. The autopilot should be able to receive a set of waypoints from the route planner, or exceptionally also from the collision avoidance system, and navigate through the waypoints step by step by calculating heading and velocity setpoints that are sent to the motor control system. The microservice includes the functionalities mentioned in Chapter 7.2, and these functionalities will be explained in further detail in this chapter. Utilization of a Kalman filter for enhancing the precision of positioning and inertial data will be discussed as well, as accurate positioning is vital for a waypoint tracking autopilot system, and for autonomous navigation systems in general.

5.1 Description of waypoint tracking algorithm

This section describes the design and development of the control algorithm for the waypoint tracking autopilot. As mentioned in the introduction of this thesis, autonomous maritime vehicles as an area of research is relatively new, and therefore, there are only a few well-known projects focusing on autonomous vessel projects. As these projects mostly are conducted by companies, access to detailed information on these projects is very limited. Finding information on the design of these navigation systems is therefore difficult and a new algorithm was needed for use on the ÅBOAT-platform.

The requirements, inputs, and outputs for the waypoint tracking autopilot were first specified. A planned route is received in the form of waypoints in a latitude, longitude coordinate system along with a defined manoeuvring tolerance and desired velocity. Position data is received from a positioning microservice that utilizes GNSS. The system outputs consist of speed and heading setpoints along with trajectory data for a collision avoidance service. Sent data can also be displayed in a graphical user interface for precise remote monitoring of the ship. The waypoint tracking algorithm should be capable of generating a path based on received waypoints and sending heading and speed setpoints for following the generated path. Position data is used for verifying correct navigation procedures and for calculating heading setpoints. The algorithm should additionally allow for smooth deceleration at the end of the route.

Different calculation functions are needed in the algorithm to achieve the above-mentioned functions. Firstly, path generation is needed for allowing more precise navigation. As mentioned earlier, different interpolation methods can be used for generating the path. The more detailed path is only generated for the current section of the route at a time to minimize memory allocation. The interpolation results in a continuous function that is discretised through calculation of the function values at certain intervals. These function values form intermediate waypoints which together form a more specific path.

As the trajectory is generated, a method for computing the desired heading is needed. This desired heading is defined as the compass bearing between the vessel's current position and the next intermediate waypoint. The heading is continuously updated and sent as the vessel moves in the coordinate system. Speed setpoints are forwarded from the planned route data to the steering and thrust control system.

A distance calculation method is also needed for obtaining the distance to waypoints and monitoring process. A waypoint is considered reached when the vessel is located within a specific radius around the waypoint, i.e., when the distance to the waypoint is less than the specified distance. Navigation continues towards the next waypoint after a waypoint is reached. Whenever a waypoint defined by the route planner is reached, a new interpolation of a route section is executed, generating new intermediate waypoints to follow.

The algorithm described above is used for manoeuvring until the last waypoint is reached and additionally slows down for the last waypoint and stops the vessel. The algorithm was initially tested using simulated positioning data and different test routes with the algorithm programmed in Matlab. Different graphical representation tools included in Matlab were used for validating the results. Simulated position data and interpolated paths were plotted on a map while heading outputs and progress data were displayed and inspected.

5.2 Route interpolation

Tracking waypoints in the form they are received from a route planner is possible in optimal weather conditions. Manoeuvring in the presence of wind, waves, and underwater streams, however, may cause a vessel to drift out of course if the distance between waypoints is long. Adding additional waypoints in shorter intervals is a solution to drifting, as it makes the route more defined. Interpolation of the route is a feasible option for generation of additional waypoints and can be done with several methods. The simplest method of generating additional waypoints is by using linear interpolation between two waypoints at a time, which results in clearly defined straight paths between waypoints. Linear interpolation is implemented in the waypoint tracking autopilot on the ÅBOAT-platform and works well for adding precision to the navigation. Although linear interpolation adds precision, it still preserves the shape of the planned route. These planned routes often include sharp turns, which in some cases can result in slightly abrupt heading transitions between waypoints. The sharp turns may reduce ride comfort and sometimes also manoeuvring efficiency. This problem is solved by choosing a smoothing interpolation method for the path generation, which generates both accurate and smooth trajectories that provide higher comfort and lower required yaw rates on a vessel. The following chapter discusses possible interpolation methods for achieving the above-mentioned features.

Three different interpolation routines are being investigated in this thesis: cubic spline, piecewise cubic hermite interpolating package, PCHIP by Fritsch (Fritsch, 1982) and a modified interpolation method by Akima (Akima, 1970). Functions for these routines are included in Matlab, which makes it easy to implement and compare these methods in experiments, as Matlab offers convenient tools for graphical representation and validation of results. All three methods rely on fitting third-order polynomial functions for each interval between connecting points. These polynomial functions are chosen so that they together create a continuous function through all the interval endpoints and so that the first derivative of the function always is continuous. Cubic spline interpolation also requires the second derivative to be continuous (Hall, Meyer, 1976), whereas this not is a requirement for the cubic spline- or modified Akima interpolation methods. As a result, cubic spline provides a smoother interpolated curve, whilst the other two methods perform better at preserving the shape of the given data for non-oscillatory data. Preserving the shape is an important factor in trajectory generation as it results in shorter routes that follow the fairways more precisely. For tight manoeuvring paths, the smooth interpolation of cubic spline interpolation can have disastrous outcomes, where the generated paths deviate significantly from the desired paths on the fairway.

The undulation that often is present in cubic spline interpolation ought to be minimized in the two latter methods, PCHIP and modified Akima (Fritsch, 1982), (Akima, 1991). A drawback in the more accurate shape preserving of PCHIP are the slightly more aggressive transitions in the connection points, which can be seen in figure 7. The modified Akima interpolation method is capable of handling fast transitions in the connecting points and provides smooth curves under most circumstances. Figures 6 and 7 show comparisons of the three interpolation methods where path trajectories between given sets of waypoints are plotted on a map. The comparisons are conducted using the functions spline, pchip and makima in Matlab. The waypoint set used in Figure 6 connects a simple route and all three interpolation methods show good and almost equal performance.

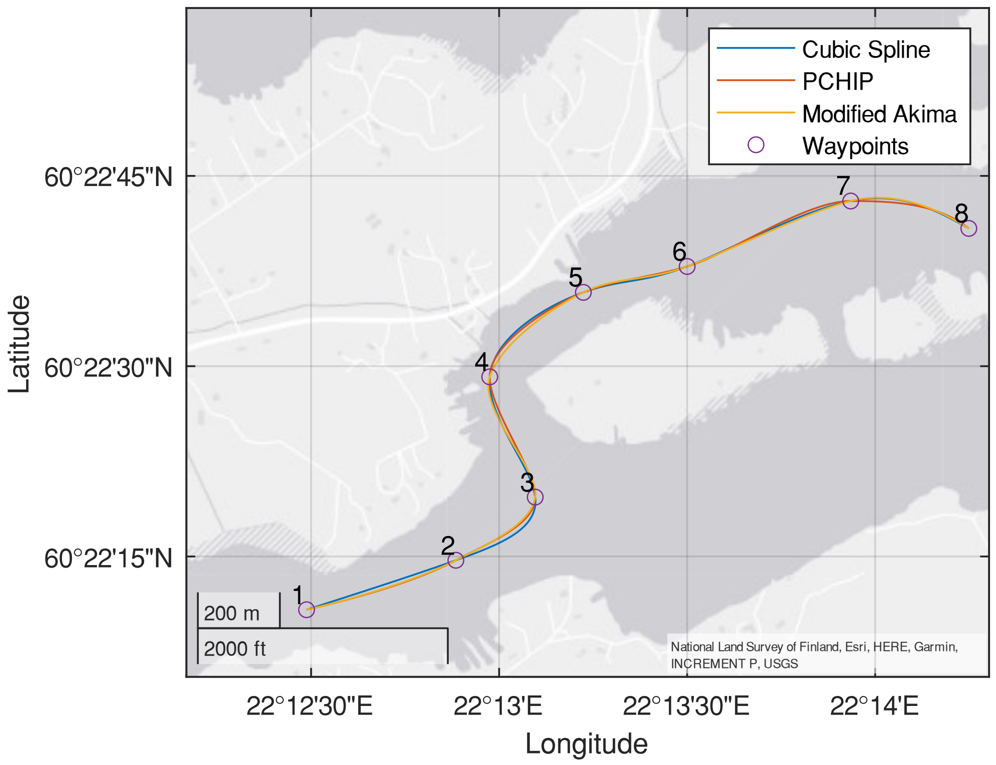


Figure 6: Comparison of interpolation methods on a simple route

Figure 7 highlights the need for an appropriate interpolation method, as the set of waypoints is chosen so that only slight overshoot and undulation is allowed. The use of cubic spline interpolation on this route is inappropriate, as the interpolated trajectory partly crosses the islands at two points and would result in a collision. Both PCHIP and the modified Akima interpolation method provide acceptable and similar route options on this more challenging route, and the most noticeable difference between the two methods is that the modified Akima interpolation provides smoother transitions in connecting waypoints two and three.

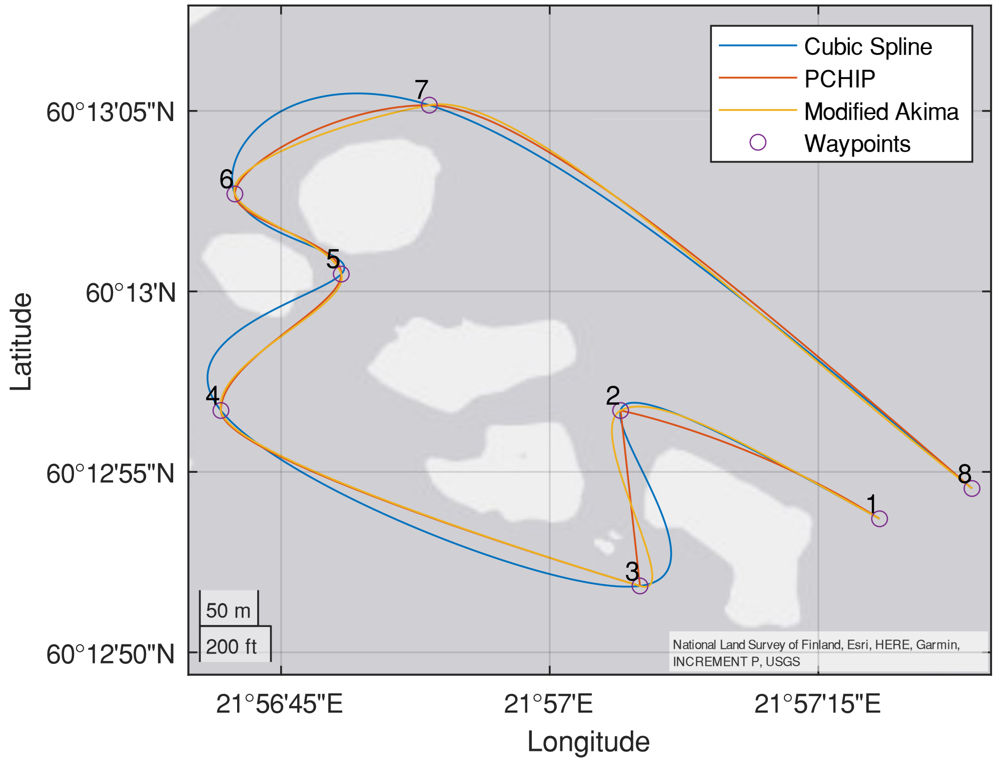


Figure 7. Comparison of interpolation methods on an advanced route

Although the modified Akima interpolation method performs well on the above comparisons, it shows slight issues when waypoints are placed close to each other. The problem can be seen in figure 8 where the modified Akima interpolation method creates a curve with unnecessary oscillation between waypoints two and three compared to the straighter and more efficient path produced by PCHIP. As figure 7 depicts a rather unusual case and more waypoints eventually would be given in tight manoeuvring situations, the benefits in smoothness and comfort of using modified Akima interpolation over PCHIP interpolation are negligible. PCHIP performs well when efficient and accurate manoeuvring on the fairway is prioritized and is therefore the chosen interpolation algorithm for smooth path generation for the ÅBOAT-project.

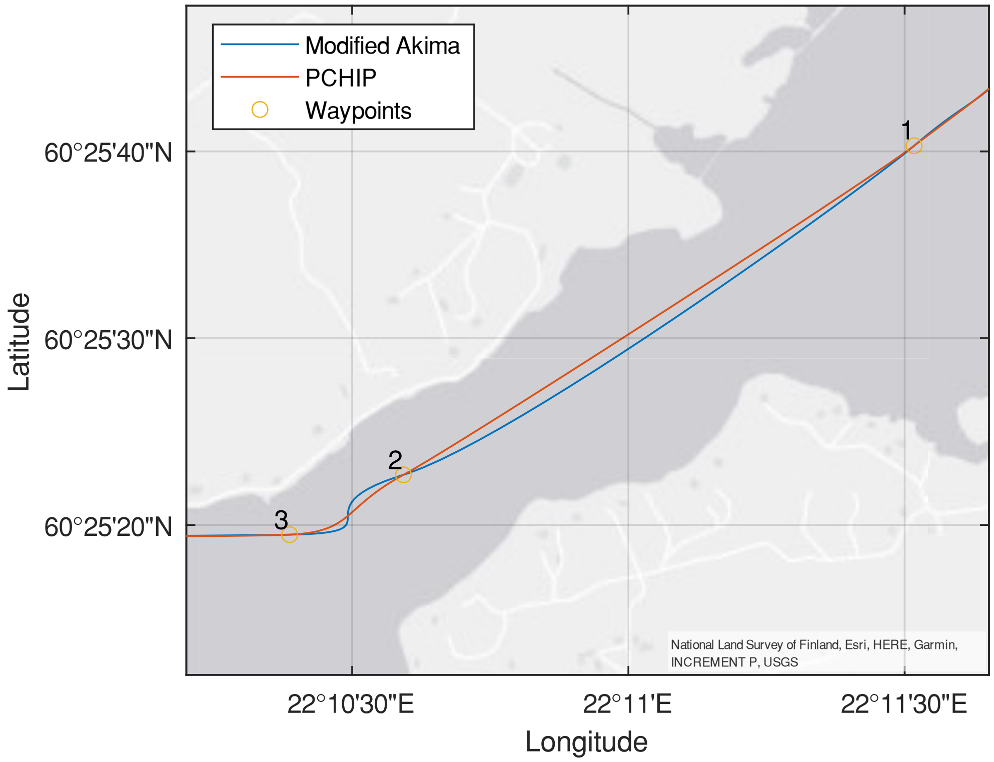


Figure 8. Comparison of modified Akima and PCHIP

As mentioned earlier, PCHIP is an interpolation algorithm which interpolates a continuous function between a set of data points by generating third-order polynomial functions specified in the Hermite form. Cubic polynomials in Hermite form are defined by their function values and their respective interval endpoints (Kreyszig, 2011). The PCHIP algorithm was originally released as a Fortran-package but is now a tool that is included in libraries for many coding languages, including Matlab and C++.

After the route has been interpolated, obtaining intermediate waypoints between the waypoints given by the route planner is possible. The number of intermediate waypoints in a certain interval can be altered to suite the complexity of the path. In the ÅBOAT-project, the number of intermediate waypoints is based on a user defined interval between the intermediate waypoints. The length of the planned route is then used for calculating the exact number of waypoints needed to achieve the desired distance between the added intermediate waypoints. This waypoint density can be tuned for achieving smooth and accurate manoeuvring and suitable values for the ÅBOAT-project were obtained through different on water tests that are described later in this thesis. The intermediate waypoints are generated by using the piecewise polynomials that are obtained with the interpolation algorithms.

5.3 Ship heading

The main manoeuvring command sent by the tracking autopilot is the desired course over ground (COG) or true ship heading. The command is generated by continuously calculating the compass bearing between the current position of the vessel and the intermediate waypoints generated by the route interpolation algorithm. The following method is used to calculate the bearing

$$\theta = \operatorname{atan2}(\sin \Delta\lambda \cdot \cos \varphi_2, \cos \varphi_1 \cdot \sin \varphi_2 - \sin \varphi_1 \cdot \cos \varphi_2 \cdot \cos \Delta\lambda),$$

where

$$\Delta\lambda = \lambda_2 - \lambda_1,$$

$$\operatorname{atan2}(y, x) = \begin{cases} \arctan\left(\frac{y}{x}\right) & \text{if } x > 0 \\ \arctan\left(\frac{y}{x}\right) + \pi & \text{if } x < 0 \text{ and } y \geq 0, \\ \arctan\left(\frac{y}{x}\right) - \pi & \text{if } x < 0 \text{ and } y < 0, \\ \frac{\pi}{2} & \text{if } x = 0 \text{ and } y > 0, \\ -\frac{\pi}{2} & \text{if } x = 0 \text{ and } y < 0, \\ \text{undefined} & \text{if } x = 0 \text{ and } y = 0. \end{cases}$$

φ_1, λ_1 are the latitude and longitude coordinates of the ship and φ_2, λ_2 are the latitude and longitude coordinates of the next intermediate waypoint given in radians. The method returns the bearing in radians. Conversion of the bearing to degrees is performed and the heading setpoint value is sent to the control system. As the atan2 function only returns values in the range -π to π, or -180° to 180°, the output must be altered further to suit the zero to 360-degree scale which is generally used for describing compass bearing.

5.4 Distance to waypoint

The tracking autopilot system is programmed to always manoeuvre towards the next intermediate waypoint until the waypoint is reached and then continue towards the next intermediate waypoint. A waypoint is considered reached when the vessel is located within a certain radius around the waypoint. This radius or distance can be altered depending on the operating environment. The distance is shorter on narrow fairways and longer on wide manoeuvring paths. The distance d between the vessel and the next waypoint is calculated using the haversine formula (Chopde, Nichat, 2013).

$$d = 2\mathrm{r}\sin^{-1}\left(\sqrt{\sin^2\left(\frac{\varphi_2 - \varphi_1}{2}\right) + \cos(\varphi_1)\cos(\varphi_2)\sin^2\left(\frac{\lambda_2 - \lambda_1}{2}\right)}\right)$$

where r is the radius of the Earth.

5.5 Kalman filter for position prediction

The waypoint tracking autopilot relies on data acquired via satellite navigation systems for manoeuvring the vessel. These systems have greatly improved and offer great accuracy, especially GNSS in combination with RTK-technology (Real Time Kinematics), which can determine a global position with a precision of only a few centimetres (Bakuła, Oszczak & Pelc-Mieczkowska, 2009). However, satellite navigation systems require an unobstructed view of the sky for optimum functionality, which cannot always be guaranteed when navigating a vessel. Problematic sections can be, for instance, bridges or tunnels where the GNSS-signal can be temporarily lost. Signal noise and low update frequency is another issue with satellite navigation systems and occurs even under conditions with stable satellite connections.

An alternative solution for predicting the position of the vessel is needed in the situations, where satellite signal is lost, and noise filtering is beneficial under all circumstances. Fossen suggests the use of a Kalman filter for predicting the position using only velocity measurements (Fossen, 2011), which can be obtained from an IMU or other velocity sensor. In addition to state prediction without GNSS-signal, Kalman filtering can be used for filtering measurement noise as well as for improving the update frequency and response of position data.

A Kalman filter is theoretically an estimator for the so-called linear quadratic problem, which is a state-estimation problem that applies to linear dynamic systems.

Measurements that are linearly related to the state are used for state prediction, resulting in a statistically optimal estimator. The computational procedure of a discrete-time Kalman estimator is described below as in the literature (Grewal, Andrews, 2001).

The procedure applies to a linear dynamic system with the discrete time model

$$\boldsymbol{x_k} = \boldsymbol{F}\boldsymbol{x}_{k-1} + \boldsymbol{G}\boldsymbol{u_{k-1}} + \boldsymbol{w_{k-1}}$$

$$\boldsymbol{z_k} = \boldsymbol{H}\boldsymbol{x}_k + \boldsymbol{v}_k \ ,$$

where $\boldsymbol{u}$ represents control inputs, $\boldsymbol{w}$ represents random dynamic disturbances, $\boldsymbol{v}$ represents random sensor noise, and $\boldsymbol{z}$ represents sensor outputs. Matrices $\boldsymbol{F}$ and $\boldsymbol{G}$ are parameters of the true system dynamic model, and matrix $\boldsymbol{H}$ contains parameters of the sensor model. The covariance matrices are defined as

$$\boldsymbol{Q} = \mathrm{E}(\boldsymbol{w}_k \cdot \boldsymbol{w}_k^T),$$

which is the covariance of process noise in the system state dynamics and

$$\boldsymbol{O} = \mathrm{E}(\boldsymbol{v}_k \cdot \boldsymbol{v}_k^T),$$

which is the covariance matrix of measurement uncertainty. $\mathrm{E}(x)$ is the Gaussian expectation operator. The computational procedure of a Kalman estimator can be divided into four steps and begins with computation of the predicted covariance matrix of state estimation uncertainty

$$\boldsymbol{P}_k(-) = \boldsymbol{F}_{k-1} \cdot \boldsymbol{P}_{k-1}(+) \cdot \boldsymbol{F}_{k-1}^{\boldsymbol{T}} + \boldsymbol{Q}_{k-1},$$

where $\boldsymbol{F}_{k-1}$ is the state transition matrix of the system and $\boldsymbol{P}_{k-1}(+)$ is the covariance matrix of state estimation uncertainty of the previous time step. The Kalman gain matrix is then obtained with the expression

$$\overline{\boldsymbol{K}}_k = \boldsymbol{P}_k(-) \cdot \boldsymbol{H}_k^T \cdot [\boldsymbol{H}_k \cdot \boldsymbol{P}_k(-) \cdot \boldsymbol{H}_k^T + \boldsymbol{O}_k]^{-1},$$

where $\boldsymbol{H}_k$ is the measurement sensitivity matrix. A correction of the state estimation covariance is obtained using $\boldsymbol{P}_k(-)$ and $\overline{\boldsymbol{K}}_k$ in the following expression,

$$\boldsymbol{P}_k(+) = (\boldsymbol{I} - \overline{\boldsymbol{K}}_k \cdot \boldsymbol{H}_k)\boldsymbol{P}_k(-).$$

As a final step, successive values of the state vector $\hat{\boldsymbol{x}}_k(+)$ are computed using the Kalman gain matrix $\overline{\boldsymbol{K}}_k$, the initial state vector $\hat{\boldsymbol{x}}_0$, and the measurement vector $\boldsymbol{z}_k$ in the expression

$$\hat{\boldsymbol{x}}_k(+) = \hat{\boldsymbol{x}}_k(-) + \overline{\boldsymbol{K}}_k[\boldsymbol{z}_k - \boldsymbol{H}_k \cdot \hat{\boldsymbol{x}}_k(-)].$$

The implementation of a Kalman filter on the ÅBOAT-platform is integrated in a separate positioning microservice which utilizes GNSS, IMU and compass sensor fusion for improved positioning performance under challenging circumstances. The positioning microservice provides filtered real-time position data for the waypoint tracking autopilot and for all other microservices on the ÅBOAT-platform, which take advantage of positioning data.

5.6 Waypoint tracking autopilot microservice system integration

Modularity is highly prioritized in the ÅBOAT project, and all the different functional modules are therefore separated as different microservices running either on hardware on the boat or on the shore control centre. As mentioned earlier, OpenDLV is used as the microservice framework on the platform. The development and initial testing of the waypoint tracking algorithm was done using Matlab, as it offered useful methods for visualizing the results with plots and other graphical presentation tools. The final software implemented on the boat, however, was developed using the coding language C++, to suit the OpenDLV framework, which is written in C++. The PCHIP interpolation method was found included in the header-only Boost library for C++ and could therefore easily be implemented in the microservice.

The OpenDLV microservices communicate with each other through OpenDLV-messages sent via User Datagram Protocol, UDP. UDP enables low latency data transfer and allows multiple microservices to receive data simultaneously. All OpenDLV-microservices acquire specific session IDs when started, and messages sent over a particular session can be received by all microservices with identical IDs without any specific addressing. Many of the messages in the session can be useful for multiple microservices, which further emphasizes the conveniency of UDP that is utilized by the OpenDLV-framework.

The waypoint tracking autopilot microservice operates according to the algorithm described in Chapter 5.1 and navigates through the waypoints step by step. The current position data are sent from the GNSS in the LTE-router to a OpenDLV-service which forwards the data to the entire session. The setpoints for the control service generated by the waypoint tracking autopilot are sent as OpenDLV-messages and can be accessed by all services in the session. Additional OpenDLV-outputs such as e.g., information on the current trajectory or remaining distance to the next waypoint, can further be configured if needed in a user interface.

6 Testing of the ÅBOAT autopilot system

Testing is a central part of developing products, and this also applies to the autopilot system on the ÅBOAT-platform. Software tests were conducted both in laboratory and on water. The algorithms in the waypoint-tracking autopilot software were first tested using graphical presentation tools in Matlab and simulated positioning data. These simulated tests verified that the output of the waypoint tracking autopilot service was correct and that the calculations worked properly. Outdoor tests on land were performed as well by carrying part of the hardware along a short route and moving in the directions calculated by the waypoint tracking autopilot. These on-land tests further incorporated the GNSS hardware and confirmed that the waypoint tracking autopilot functioned properly in actual use. The real behaviour of the complete autopilot system is, however, difficult to test in a laboratory environment, and on-water tests were therefore needed to tune the system.

The on-water tests are vital for testing the functionalities of all systems onboard the ÅBOAT-platform. Accurate sensor data reading and reliable data transfer and logging are critical for the development of the autopilot system. Several test scenarios were planned both for testing the functionalities of the waypoint tracking system as well as for tuning the response of the steering and thrust control system. The steering- and thrust control system was initially tuned by navigating on simple, but demanding routes that set high requirements on the response of the steering and thrust control system.

6.1 Test setup

Several on-water tests of the ÅBOAT-platform were performed in Ruissalo, Turku, where different routes were pre-planned for the waypoint tracking autopilot. The planned routes gradually increase in length and difficulty, and the first two routes are mainly designed for testing and tuning the PID-controllers in the steering and thrust control system. The 90-degree turns in the first route provide good opportunities for setting up the controllers to generate reasonable responses for abrupt changes in heading setpoints. The first test route is pictured in figure 9.

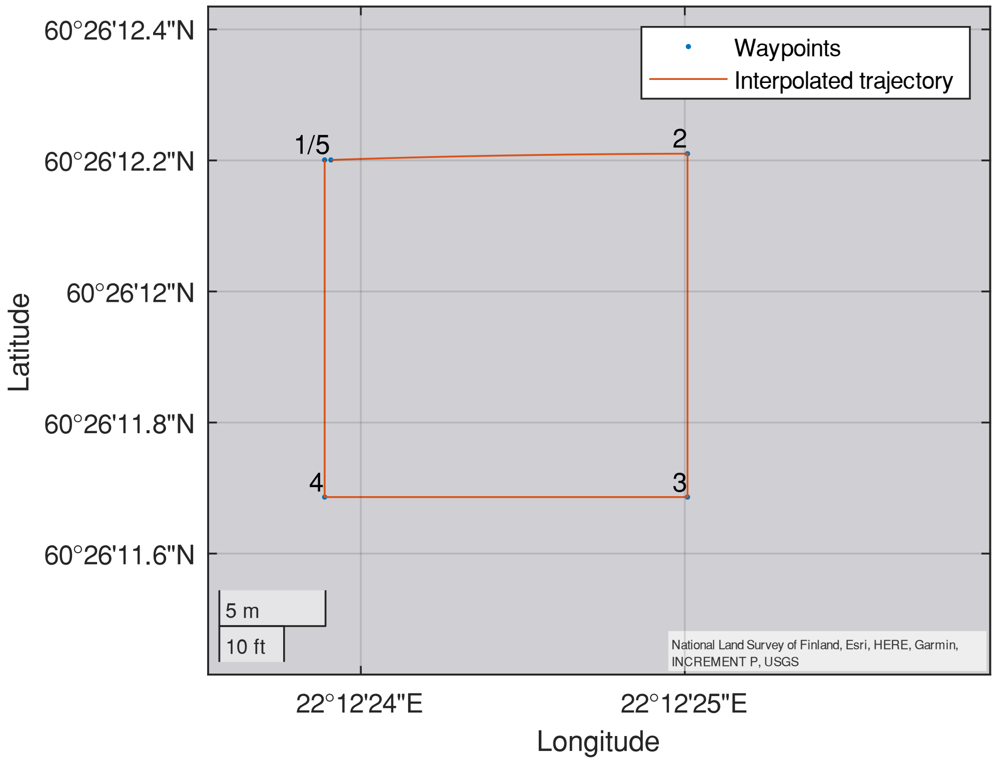


Figure 9. Test route 1.

The second route, pictured in figure 10, includes even sharper turns which sets higher requirements on the steering and thrust control system. Linear route interpolation is best suited for testing on the first two routes due to their short lengths and due to the purpose of the tests. Linear route interpolation retains the sharp turns and drastic setpoint changes which is preferred for PID parameter tuning.

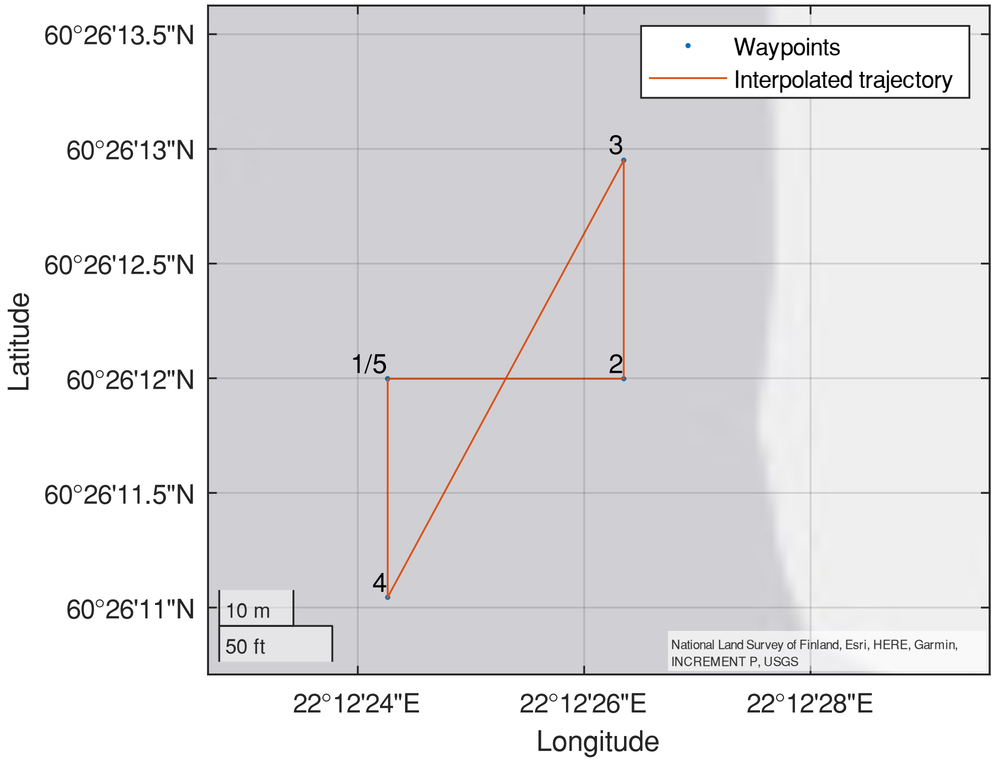


Figure 10. Test route 2.

Two longer test routes with a higher number of waypoints are planned at the same location as well. These routes test the robustness of the system in prolonged use and allows for testing the smooth path generation mode of the waypoint tracking autopilot. Test route 3 is 308 metres long and consists of six waypoints, whereas test route 4 is 500 metres long and consists of 11 waypoints. The waypoints along with hypothetical smoothly interpolated trajectories for test routes 3 and 4 are pictured below in figures 11 and 12.

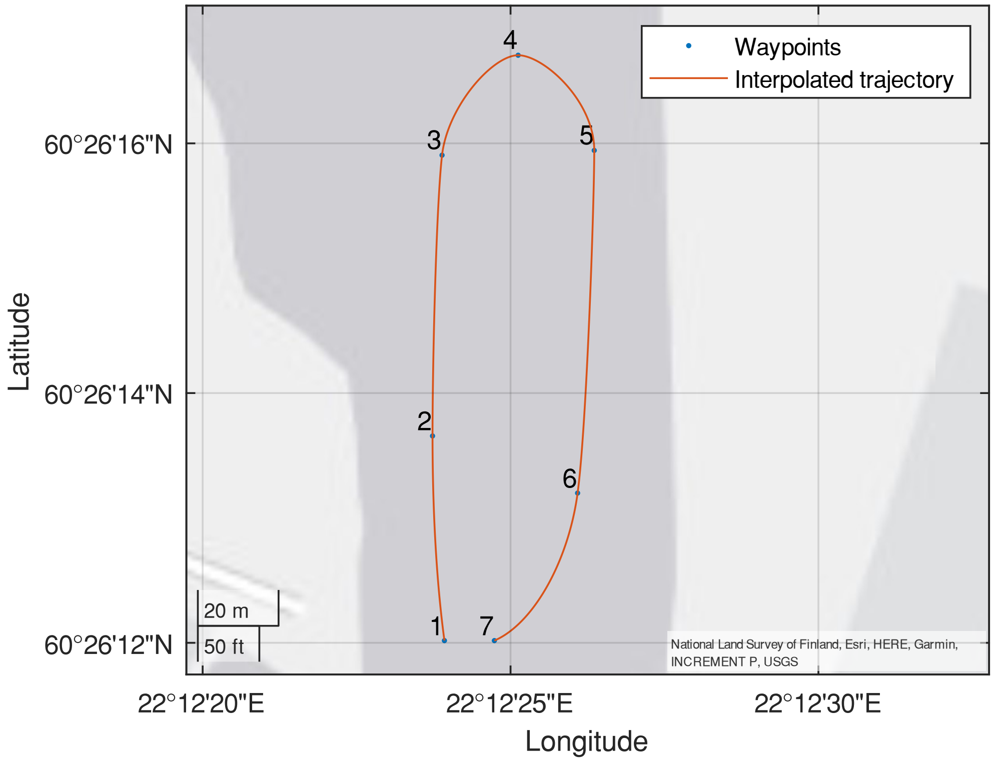


Figure 11. Test route 3.

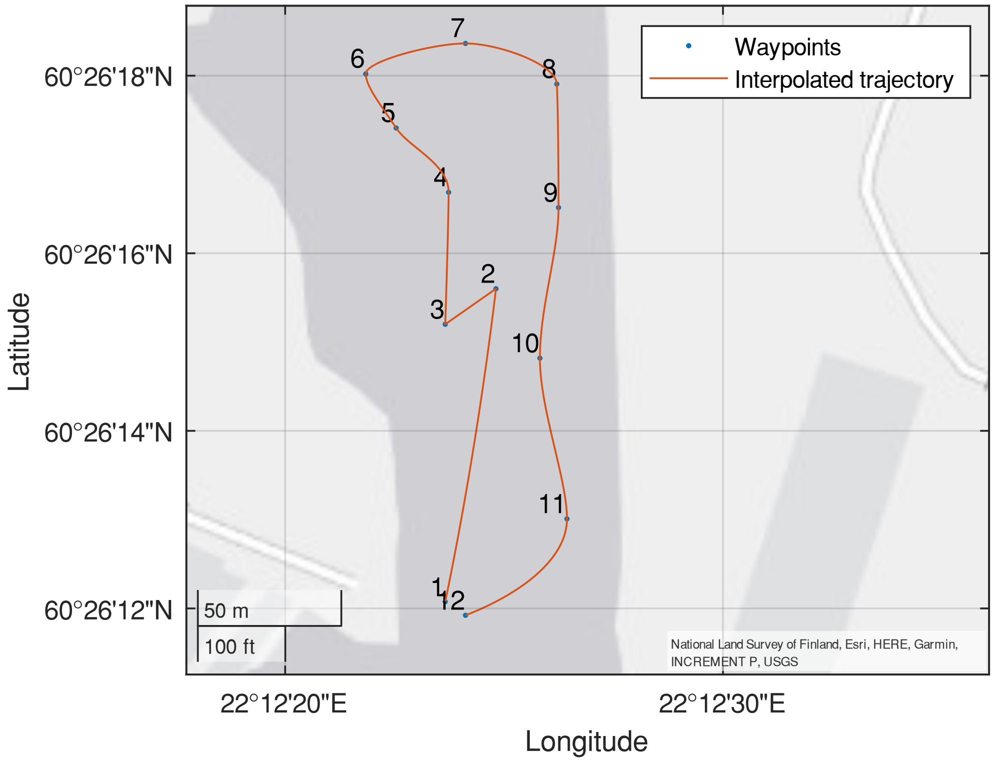


Figure 12. Test route 4.

6.2 Results

On-water tests on the previously mentioned routes were conducted throughout the writing of this thesis. Varying results were attained as the platform still was under development. The main difficulties during the tests were in the generation of reliable sensor data for the platform's movement in surge- and sway directions, which resulted in difficulties in steering- and thrust control on the platform. However, several tests were conducted on test route 1 and 3 where the platform managed to complete the route, which confirmed that the algorithm of the waypoint tracking autopilot operated correctly. Tests on routes 2 and 4 are to be performed later, when the positioning system and IMU sensor systems of the platform are further developed. The best results were achieved using only two of the PID controllers for controlling the surge velocity and yaw while inactivating the control of the platform's sway velocity.

The tests on route 1 were performed with the smoothing interpolation deactivated as the main goal was to evaluate the basic functionalities of the system. The waypoint radius was also set fairly high to 5 metres as the manoeuvring accuracy was restricted by the insufficient sensor data and mediocre controllability. After tuning the PID controllers, the platform managed to manoeuvre autonomously along the route, reaching all five waypoints within the set margin of five metres. Plotted position data from a test where the ÅBOAT platform is manoeuvred on route 1 is pictured below.

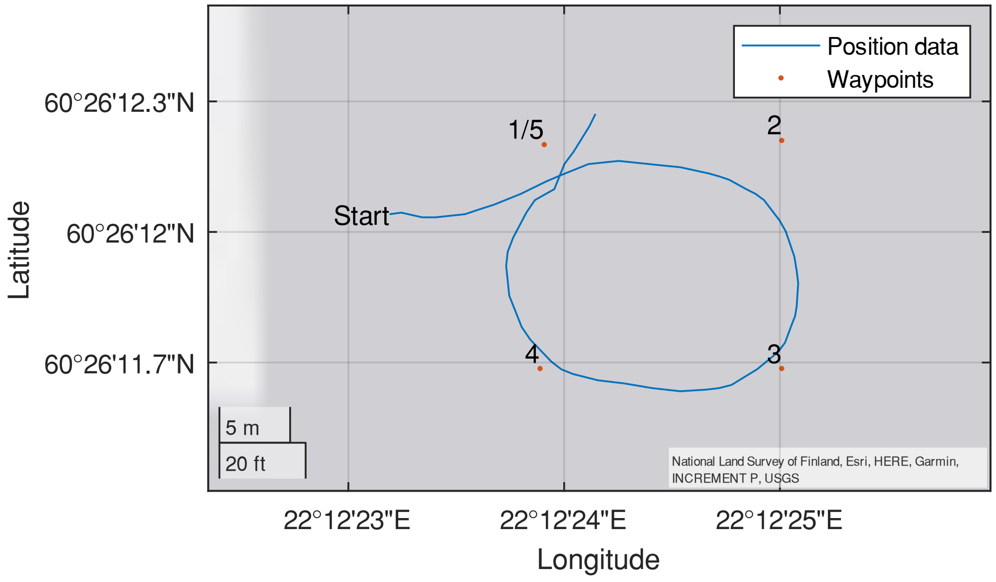


Figure 13. Results from test on route 1

Several successful attempts were executed on test route 3, where the smooth interpolation mode of the waypoint-tracking autopilot could be tested. The waypoint radius was set to 5 metres on these tests as well, which resulted in flawless navigation without any circling around the waypoints. The smooth trajectory generation function of the waypoint tracking system was proven to be functional with the tests, although the benefits are negligible on short routes as the interval between intermediate waypoints would have to be set shorter in order to generate a completely smooth path. The intermediate waypoint density was set to 5 intermediate waypoints per 100 metres during all the tests, which only resulted in one to five added intermediate waypoints between waypoints on the test routes presented in Chapter 6.1. and therefore, only resulted in minor smoothing of the route. Figure 14 presents the results of a successful test on route 3 where the above-mentioned parameters are used.

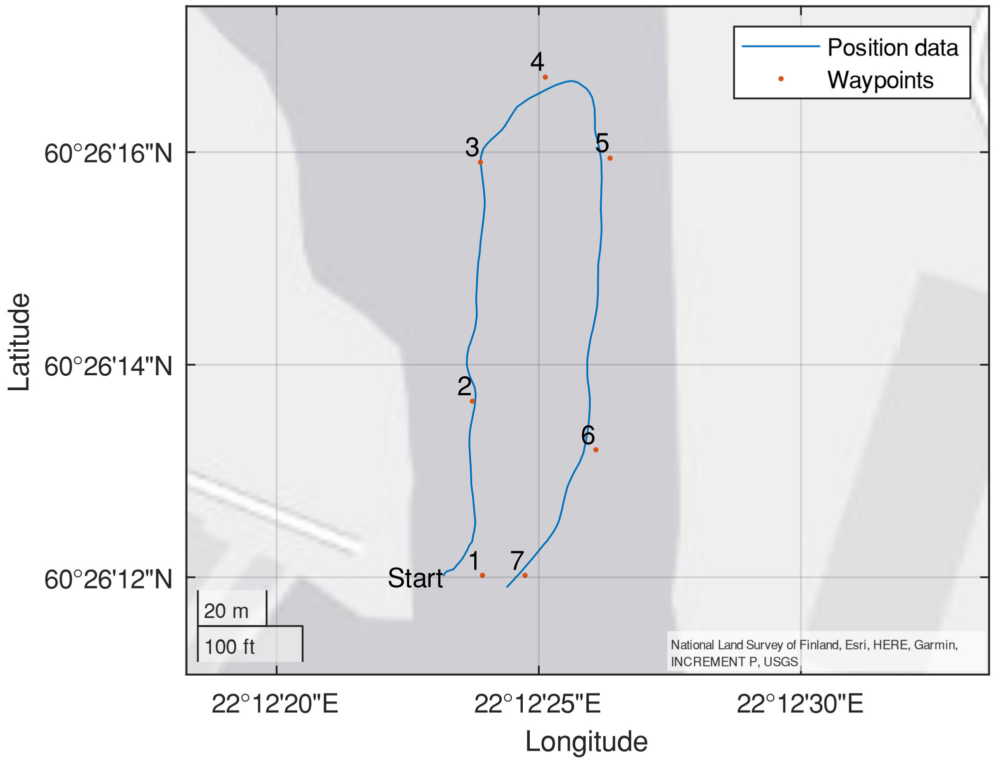


Figure 14. Results from test on route 3

Although the system functioned well under the tests pictured above, there were still numerous occasions where the autonomous navigation system failed to complete the routes and began circling around waypoints. The reasons for failure were generally problems in the control system due to insufficient or fluctuating sensor data. The mentioned issues lead to poor performance, especially when navigating in the presence of slightly higher wind speeds or in currents. Poor software system reliability was another common issue which led to failure, although it is acceptable at this early stage of development. A typical scenario where the control system fails is pictured in figure 15.

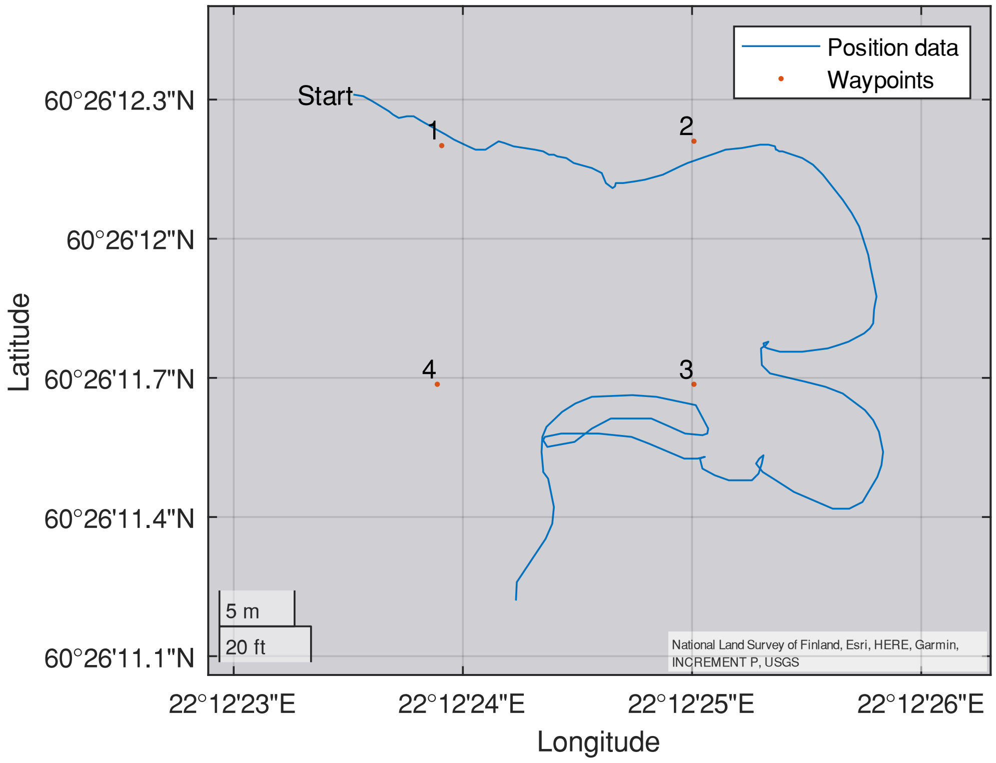


Figure 15. Uncompleted test on route 1

Overall, the presented results prove the concept of the autonomous navigation system on the ÅBOAT-platform and shows that its main functionalities are working. In addition, the acquired results also provided useful information and insights on current limitations of the system. As previously mentioned, improved accuracy of the inertial data is needed to achieve proper control of steering and thrust, which would enable shorter waypoint radius and more precise manoeuvring. Further development of the ÅBOAT platform is therefore planned to be conducted on sensor fusion between the IMU and GNSS to achieve the needed improvements.

Another finding was that the advantages of smooth route interpolation for the ÅBOAT platform on the test routes seem to be negligible in terms of comfort and economy when the vessel only travels at low speeds. It does, however, assist the steering and thrust control system by providing smoother transitions between heading setpoints. It is easier to tune the PID-parameters for these smoother transitions and overshoot is less likely as well. The smooth route interpolation feature may, however, be more useful when used on other vessels where it can improve comfort and economy as well, and perhaps also on an updated version of the ÅBOAT-platform traveling on longer routes at higher speed.

7 Summary

This thesis has discussed autonomous navigation systems, with focus on their design and possible areas of application. The thesis has covered the fundamentals of the autonomous navigation system design for the ÅBOAT-platform, with particular focus on the development of the waypoint tracking autopilot system. Descriptions and results of tests performed with both the waypoint tracking autopilot system on its own, and the complete autonomous system on the ÅBOAT were also included in the thesis.

Finding an appropriate method for path generation was the main challenge in the development of the waypoint tracking autopilot. Three different interpolation methods for generation of smooth paths in the waypoint tracking autopilot have been investigated and compared. The compared interpolation methods were cubic spline interpolation, PCHIP, and modified Akima interpolation. The comparison showed that cubic spline interpolation produced the smoothest paths but lacked precision in challenging conditions. Modified Akima interpolation performed well in both preserving shapes and smoothing curves but generated somewhat inefficient paths in some certain cases where the waypoints are located too close to each other. PCHIP interpolation performed well at preserving the shape of datapoints and thereby generated both efficient and precise paths. The paths generated by PCHIP were mainly smooth, but included sharper turns compared to modified Akima interpolation in some scenarios. PCHIP still seemed to perform the best and was therefore chosen as the smoothing path generation method for the waypoint tracking autopilot.

The thesis covered the design of other components and functions in the waypoint tracking autopilot as well. Methods for calculating distance to waypoints and compass bearing between coordinates in the world reference frame were described. The thesis also included a short description of the theory behind Kalman filtering, which in maritime applications can be used for achieving more responsive and accurate data for describing the movement and position of a vessel.

8 Summary in Swedish – Svensk sammanfattning

Utveckling av autonoma navigationssystem för vattenburna fordon

Introduktion

Under det senaste decenniet har intresset för autonom teknik inom transportindustrin ökat markant. Omfattande forskning och utveckling av självgående apparater bedrivs inom många branscher, men speciellt synligt är fenomenet inom bilbranschen. Möjligheterna till autonom teknik utforskas även i den maritima industrin och det finns många eventuella fördelar med autonom sjöfart.

Syftet med detta diplomarbete är att studera autonoma maritima navigationssystem och utveckla programvara för ÅBOAT-projektet vid Institutionen för informationsteknologi vid Åbo Akademi. Autonoma maritima navigationssystem diskuteras som en helhet och de mest centrala funktionerna beskrivs på en allmän nivå. Det primära målet med arbetet är att utveckla ett autopilotsystem som möjliggör navigering längs en rutt som är definierad i delmål beskrivna med koordinater (eng. *waypoints*). Autopilotsystemets huvudsakliga funktioner är att anpassa lämpliga och väldefinierade banor till de givna delmålen och att på basen av positioneringsdata och kartor tillhandahålla ett motorreglersystem med börvärden för kurs och hastighet för att manövrera farkosten längs en rutt via de givna delmålen. Utvecklingen av motorernas reglersystem och positioneringssystemet för ÅBOAT-plattformen diskuteras i detalj, eftersom autopilotsystemet är beroende av deras funktioner.

Funktionsprincipen för autonoma maritima navigationssystem

Autonoma navigationssystem kan vara konstruerade på många olika sätt, men i den här avhandlingen betraktas autonoma navigationssystem som fyrdelade, bestående av ruttplanerare, autopilotsystem, motorreglersystem samt ett system för situationsmedvetenhet. Komponenterna bildar tillsammans ett system som kan manövrera ett fartyg autonomt från en plats till en annan. Ruttplaneraren definierar en rutt i form av delmål och autopilotsystemet genererar banor för rutten och beräknar kurs- och hastighetsbörvärden som sänds till motorreglersystemet.

Motorreglersystemet omvandlar i sin tur kurs- och hastighetsbörvärden till faktiska styrsignaler för motorerna. Systemet för situationsmedvetenhet använder sig bland annat av kameror med objektigenkänning, LIDAR-system, GNSS och AIS för att noggrant bestämma tillståndet för farkosten och dess omgivning. Detta system samverkar med autopilotsystemet och korrigerar vid behov rutten för att undvika kollision med icke-statiska hinder.

ÅBOAT-projektet

Navigations- och reglersystemen i denna avhandling är särskilt utformade för en autonom båt som utvecklas vid institutionen för informationsteknologi vid Åbo Akademi som en del av MAST! -projektet. Båten fungerar främst som en pilotplattform för att utveckla ett autonomt navigationssystem och ett mål är att utveckla ett konfigurerbart system som kan tillämpas även på andra fartyg. Plattformen ska kunna fungera helt autonomt men den ska också kunna manövreras fjärrstyrt och övervakas från en kontrollstation på land. En bred kompatibilitet uppnås genom att använda en modulär hårdvarudesign där de olika komponenterna består av allmänt tillgängliga enheter och där standardanslutningar används. Mjukvaran på ÅBOAT-plattformen är uppbyggd på OpenDLV-ramverket som är en microservice-baserad miljö anpassad för utveckling av autonoma fordon (OpenDLV, 2020). De olika delarna av autonoma navigationssystemet är således alla byggda som åtskilda tjänster och körs individuellt för ökad modularitet.

ÅBOAT-plattformen är byggd på en hopvikbar och uppblåsbar fiskebåt som är 375 cm lång. Chassits platta utformning möjliggör enkel installation av hårdvara och dess portabilitet är en fördel då båten ska transporteras mellan olika testmiljöer. Plattformen drivs med två elektriska trollingmotorer monterade i vardera ända av båten och dessa fungerar som roderpropellrar. Motorerna styrs av en Raspberry Pi enkortsdator som är kopplad till motorerna med NMEA2000-anslutningar via en CAN-styrenhet. Majoriteten av funktionerna som är relaterade till motorstyrning och autopilotsystemet körs på Raspberry Pi enkortsdatorn. Situationsmedvetenhet uppnås med hjälp av fyra kameror och en LIDAR-enhet som är kopplade till en Nvidia Jetson Xavier AI-dator som bearbetar data och identifierar omgivningen runt ÅBOAT-plattformen för att undvika kollisioner. Ett LTE-modem används för att ansluta plattformen till kontrollstationen på land. LTE-modemet tillhandahåller också ÅBOAT-systemet med information om båtens position och rörelse med hjälp av ett inbyggt GNSS-system. En kompass/IMU-enhet ger ytterligare information som behövs för regleringen av båtens kurs och hastighet. Hela systemet ombord på ÅBOAT-plattformen drivs med hjälp av en blyackumulator med en spänning på 12 V och en kapacitet på 80 Ah.

Utveckling av autopilotssystemet för ÅBOAT-plattformen

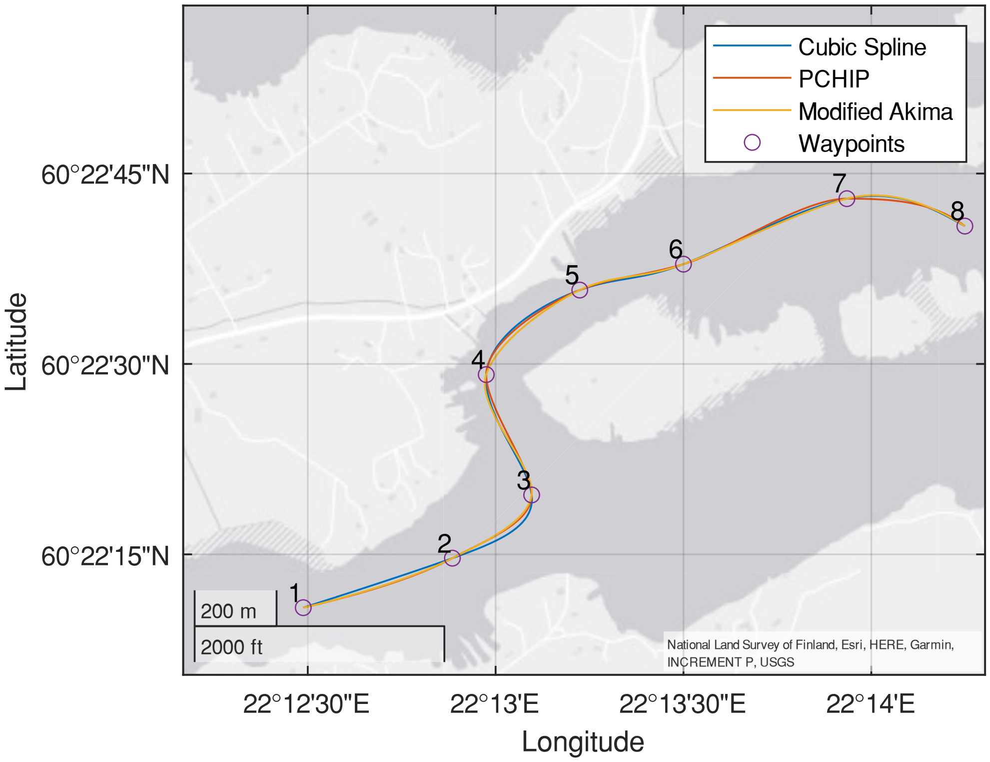
Autopilotsystemet är en central del av det autonoma navigationssystemet på ÅBOAT-plattformen. Fullständiga lösningar för autonom manövrering av fartyg finns inte allmänt tillgängliga och därför utvecklades ett nytt autopilotsystem. De huvudsakliga funktionerna i autopilotsystemet är färdbanegenerering, beräkning av avstånd mellan delmål samt beräkning av kompassriktning till delmål. Med hjälp av de ovannämnda funktionerna kan ett system skapas som kan räkna ut börvärden för kurs och hastighet för navigering längs en rutt som är definierad i delmål.

Generering av färdbana

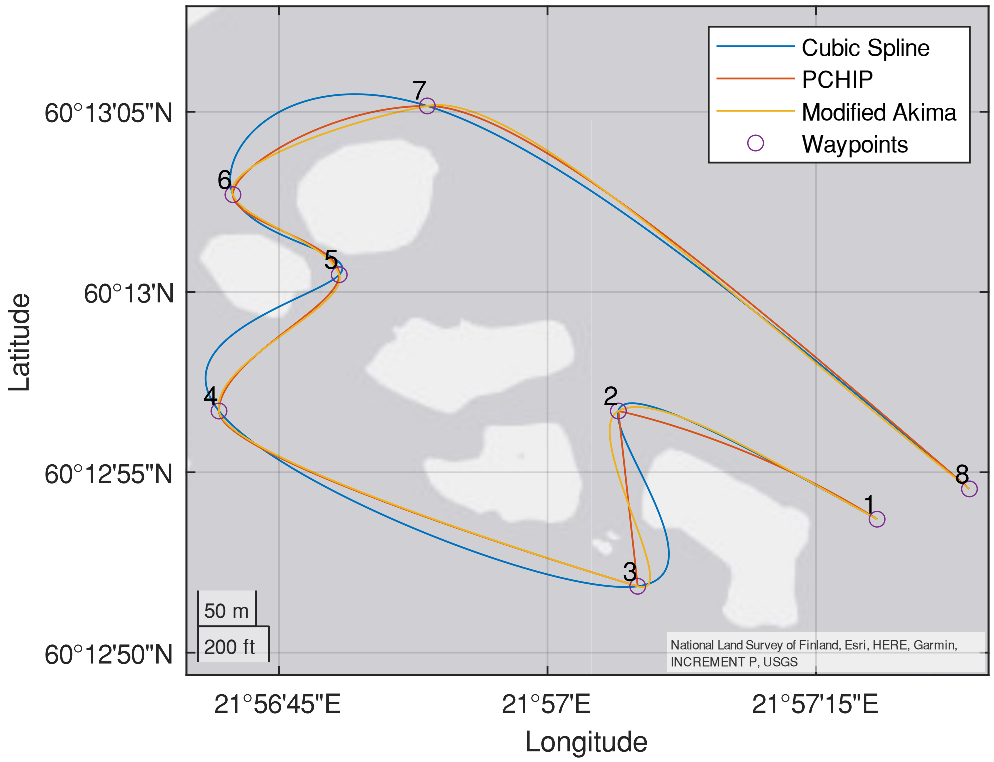
Det finns flera metoder för generering av färdbanor som går genom definierade delmål, varav den enklaste metoden är linjär interpolation. Metoden är lämpad för situationer där vikten ligger på att navigera med så stor precision som möjligt och finns därför med i autopilotsystemet på ÅBOAT-plattformen. För att utöver detta åstadkomma mer komfort och optimera effektiviteten i kurvor, krävs andra metoder. I detta arbete undersöks tre olika utjämnande interpolationsmetoder för generering av färdbanor: kubisk spline-interpolation, kubisk hermitisk interpolation och modifierad Akima-interpolation.

Dessa tre interpolationsmetoder grundar sig på anpassning av tredje ordningens polynomfunktioner för intervall mellan datapunkter. Polynomfunktionerna väljs så att de tillsammans skapar en kontinuerlig funktion genom alla datapunkter och så att polynomens första derivata alltid är kontinuerlig. Kubisk spline-interpolation kräver dessutom också en kontinuerlig andraderivata vilket resulterar i ännu mjukare övergångar mellan funktionernas knytpunkter. Funktioner för alla tre metoder finns inkluderade i programmet Matlab med namnen, *spline, pchip* respektive *makima,* vilket underlättade jämförelsen av metoderna.

För jämförelse av metodernas prestanda planerades först rutter av olika komplexitet med hjälp av en ruttplanerare i karttjänsten Openseamap. Rutternas delmålskoordinater interpolerades sedan med de olika interpolationsmetoderna i Matlab, varefter de grafiskt presenterades och jämfördes med Matlabs inbyggda ritfunktioner. Alla metoder genererade liknande banor för de enklare rutterna, vilket kan ses i figur 16, men för de mer~~a~~ komplexa rutterna fanns det klara skillnader. Kubisk spline-interpolation gav de mjukaste banorna, men banorna avvek också mest från en linjärt interpolerad rutt. Figur 17 visar hur interpolationsmetoderna beter sig för en komplex rutt, och det kan konstateras att kubisk spline-interpolation i detta fall genererar en bana som avviker från en linjärt interpolerad bana och som kan ge upphov till farliga situationer inom det planerade användningsområdet.

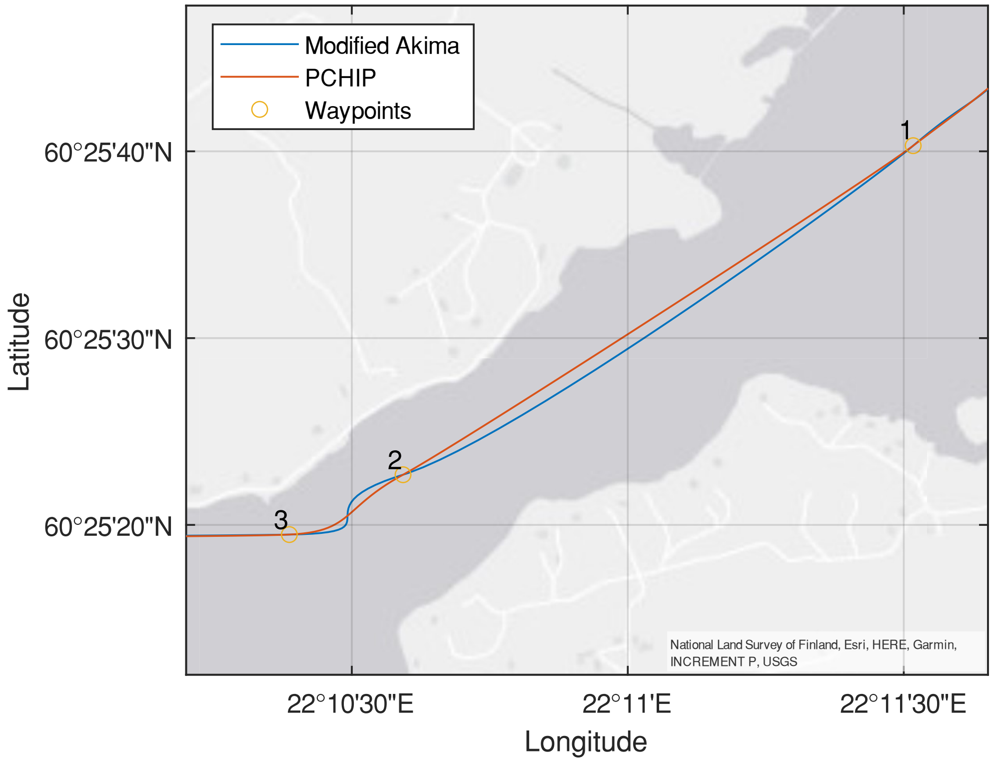


Figur 16. Interpolationsmetodernas prestanda på enkel rutt



Figur 17. Interpolationsmetodernas prestanda på komplex rutt

Både PCHIP- och modifierad Akima-interpolation gav liknande resultat som båda följde den linjärt interpolerade banan men även mjukade upp kurvorna. I snäva kurvor lyckades modifierad Akima-interpolation bättre med att utjämna kurvorna än PCHIP-interpolation. Modifierad Akima-interpolation genererar dock aningen ineffektiva banor då två delmål är placerade nära varandra i förhållande till resten av delmålen, vilket kan ses i figur 18. PCHIP-interpolation visade sig därmed vara den mest lämpade metoden för utjämnande färdbanegenerering för ÅBOAT-plattformen.



Figur 18. Jämförelse av modifierad Akima-interpolation och PCHIP-interpolation

Systemets funktioner

Förutom färdbanegenereringen har systemet också andra centrala funktioner. Banornas matematiska funktioner används för att räkna ut ytterligare mellandelmål i intervall som är specificerade av användaren. Då mellandelmålen är beräknade manövreras farkosten via alla delmål genom beräkning av kompassriktningen mellan farkostens nuvarande position och nästa mellandelmål. Autopiloten avgör ifall delmålen är nådda genom att beräkna avståndet till delmålet med hjälp av Haversine-formeln. Då ett delmål har nåtts, siktar autopiloten på nästa delmål och upprepar de ovan nämnda beräkningarna. Då sista delmålet har nåtts avslutas beräkningarna och hastighetsbörvärdet ställs till noll.

Test av ÅBOAT-plattformen

Flera olika test utfördes vid utvecklingen av autopilotsystemet för ÅBOAT-plattformen. I utvecklingens startskede testades navigeringsalgoritmerna med hjälp av att simulera positionsdata för olika rutter och granska de genererade börvärdenas riktighet. Efter att algoritmen fungerade problemfritt i de simulerade testen utfördes tester utomhus på land. Testerna utomhus utfördes genom att bära omkring en del av hårdvaran med en GNSS enhet längs en kort rutt enligt riktningarna som autopilotalgoritmen genererar och kontrollera börvärdenas riktighet.

För att optimera autopilotsystemets prestanda och funktion bör autonoma navigationssystemet testas i sin helhet vilket enklast görs på vatten. Flera tester på vatten gjordes med ÅBOAT-plattformen för att justera motorkontrollsystem, reglersystem, autopilotsystem och fjärrstyrningssystem. Fjärrstyrning testades först för att försäkra att motorstyrningen fungerade korrekt varefter autopilotsystemet och reglersystemets funktioner testades. Fyra olika rutter av olika svårighetsgrad planerades för testerna där de första rutterna är enkla och lämpar sig bra för att justera PID-reglersystemets parametrar och där de senare är lite längre och testar det autonoma navigationssystemets funktion som helhet.

Ett flertal tester på vatten utfördes parallellt med skrivandet av denna avhandling med varierande resultat och under några omständigheter lyckades plattformen navigera autonomt längs två av de planerade rutterna. Dessa test bevisade att det autonoma navigationssystemet, inklusive den utjämnande färdbanegenereringen fungerade. Under dessa test var dock manövreringsprecisionen inte så hög eftersom systemet för att mäta ÅBOAT plattformens hastighet framåt och i sidled inte ännu var färdigutvecklat, vilket i sin tur ledde till att PID reglersystemet inte fungerade som det skulle. De bästa resultaten åstadkoms då enbart kursen och hastigheten i framåtriktning reglerades och endast två PID regulatorer användes. Vid hårda vindar och vattenströmmar var det dock svårt för systemet att reglera kursen med den begränsade hastighetsregleringen eftersom motorerna inte kompenserade för de externa krafterna. Detta ledde till att plattformen började cirkla runt delmålen och misslyckades med att fullfölja rutterna.

Under dessa tester kunde det konstateras att börvärderna som genererades var korrekta och att autopilotsystemet således fungerade korrekt. Testerna gav även information om ÅBOAT plattformens brister och hur den kunde vidareutvecklas. Testresultaten belyser vikten av att ha tillförlitliga mätdata vid reglering av system och dessutom kunde det konstateras att manövreringsprecisionen för plattformen borde förbättras. Trots att fördelarna med utjämnande färdbanegenerering var marginella ur en ekonomisk synvinkel, så hade funktionen andra fördelar kopplade till regleringen. Funktionen bidrog till mjukare ändringar i börvärden som är enklare att reagera på för reglersystemet och minskade risken för översläng. De största fördelarna i komfort och effektivitet skulle dock förmodligen uppnås ifall systemet användes vid högre hastigheter på längre rutter.

Slutsats

I denna avhandling har autonoma navigationssystem diskuterats med fokus på deras design och möjliga tillämpningsområden. Avhandlingen har behandlat designen för ÅBOAT-plattformens autonoma navigationssystem, med tyngdpunkt på utvecklingen av plattformens autopilotsystem. De olika testerna som utförts vid utvecklingen av ÅBOAT-plattformens autonoma navigationssystem och dess delkomponenter har beskrivits i avhandlingen.

Att hitta en lämplig metod för utjämnande färdbanegenerering var den största utmaningen vid utvecklingen av autopilotsystemet. Tre olika interpolationsmetoder för generering av färdbanor har undersökts och jämförts: kubisk spline-interpolation, PCHIP och modifierad Akima-interpolation. Jämförelsen visade att kubisk spline-interpolation gav de mjukaste banorna men saknade precision under utmanande förhållanden. Modifierad Akima-interpolation bevarade datapunkternas former bra samtidigt som den också gav banorna en mjuk form, men i vissa fall blev de genererade banorna något ineffektiva med extra svängar. Även PCHIP-interpolation bevarade datapunkternas form och genererade effektiva och exakta banor under alla omständigheter. Banorna som genererades av PCHIP var huvudsakligen smidiga men inkluderade i vissa fall skarpare svängar jämfört med banor genererade med modifierad Akima-interpolering. PCHIP verkade på grund av sin jämna prestanda ändå passa bäst för ÅBOAT-systemet och valdes därför som metod för utjämnande generering av färdbanor i autopilotsystemet.

I avhandlingen har även autopilotsystemets andra komponenter och funktioner diskuterats. Metoder för beräkning av avstånd till delmål och beräkning av kompassriktning mellan koordinater har beskrivits. Avhandlingen innefattar också en kort beskrivning av teorin bakom Kalman-filtrering, som inom sjöfart kan användas för att uppnå bättre respons och högre precision för data som beskriver fartygets rörelse och position.

9 Bibliography

Akima, H. (1991) ‘A method of univariate interpolation that has the accuracy of a third-degree polynomial’, *ACM Transactions on Mathematical Software*, vol. 17, no. 3, pp. 341-366.

Akima, H. (1970) ‘A new method of interpolation and smooth curve fitting based on local procedures’, *Journal of the ACM*, vol. 17, no. 4, pp. 589-602.

Åström, K.J. & Källström, C.G. (1976) ‘Identification of ship steering dynamics’, *Automatica*, vol. 12, no. 1, pp. 9-22.

Bakuła, M., Oszczak, S. & Pelc-Mieczkowska, R. (2009) "Performance of RTK positioning in forest conditions: Case study", *Journal of Surveying Engineering*, vol. 135, no. 3, pp. 125-130.

Beggs, J.S. (1983) *Kinematics*. 1st edn. Washington: Hemisphere Publishing.

Business Finland. (2018) *Rolls-Royce and Finferries demonstrated world's first autonomous ferry in Finland* [Online]. Available at: https://businessfinland.fi/en/whats-new/news/2018/rolls-royce-and-finferries-demostrated-worlds-first-autonomous-ferry-in-finland (Accessed: 28 April 2021).

Chopde, N.R. & Nichat, M. (2013) ‘Landmark based shortest path detection by using A* and Haversine formula’, *International Journal of Innovative Research in Computer and Communication Engineering*, vol. 1, no. 2, pp. 298-302.

Fossen, T.I. (2011) *Handbook of marine craft hydrodynamics and motion control*. 1st edn. Chichester: John Wiley & Sons.

Fossen, T.I. (2002) *Marine Control Systems–Guidance, Navigation, and Control of Ships, Rigs and Underwater Vehicles*. 1st edn. Trondheim: Marine Cybernetics.

Fritsch, F.N. (1982) *Piecewise Cubic Hermite Interpolation Package - Final specifications*. Version 8.5, Livermore: Lawrence Livermore National Lab.

Fröjdö, L. (2021) ’Reglersystem för kurs och hastighet för autonomt fartyg’, BSc thesis, Åbo Akademi University, Turku.

Grewal, M.S. & Andrews, A.P. (2001) *Kalman filtering: Theory and Practice with MATLAB*, 2nd edn. New York: John Wiley & Sons.

Hall, C.A. & Meyer, W.W. (1976) ‘Optimal error bounds for cubic spline interpolation’, *Journal of Approximation Theory*, vol. 16, no. 2, pp. 105-122.

Häggblom, K. & Böling, J. (2013) *Reglerteknik II, Tillståndssystem*. 1st edn. Turku: Åbo Akademi University.

International Maritime Organization. (1972) *Convention on the International Regulations for Preventing Collisions at Sea, 1972 (COLREGs)* [Online]. Available at: https://www.imo.org/en/About/Conventions/Pages/COLREG.aspx (Accessed: 11 May 2021).

Kreyszig, E. (2011) *Advanced Engineering Mathematics*, 10th edn. New Jersey: John Wiley & Sons.

OpenDLV. (2020) *OpenDLV* [Online]. Available at: https://opendlv.org/ (Accessed: 26 April 2021).