NTNU
Norwegian University of
Science and Technology

Design and Implementation of Control System for Green Unmanned Surface Vehicle

Bendik Olai Agdal

Master of Science in Cybernetics and Robotics
Submission date: June 2018
Supervisor: Tor Arne Johansen, ITK

Norwegian University of Science and Technology
Department of Engineering Cybernetics

NTNU
Norwegian University of
Science and Technology

Faculty of Information Technology
and Electrical Engineering
Department of Engineering Cybernetics

MSC THESIS DESCRIPTION SHEET

Name: Bendik Olai Agdal
Department: Engineering Cybernetics
Thesis title: Design and Implementation of Control System for Green Unmanned Surface Vehicle

Thesis Description:

The project aims to design and implement an onboard control system for a wave and solar powered unmanned surface vehicle (USV). This includes design, selection of components, implementation and testing of USV.

The master project will be based on the findings in the project report that was written by the student in semester 1. The project report focused on gaining understanding of technicalities related to real-life use of USVs, specification of requirements and proposed a high level design for the system. The system specifications and high level design used in the master project is based on the solution that was proposed in the semester 1 project.

An apprentice in automation was made available to the student for selecting of connectors, cables, fuses and relays. The apprentice made the detailed schematics for the system and performed the wiring and mounting of devices in coordination with the student.

Most of the selected components arrived in the end of February, but the delivery of the USV was delayed to March 21st. At delivery, the student had engineers from AutoNaut Ltd. available to answer questions. The student was also able to request necessary information from AutoNaut via email prior to and after delivery.

Start date: 2018-01-15
Due date: 2018-06-10

Thesis performed at: Department of Engineering Cybernetics, NTNU
Supervisor: Professor Thor Arne Johansen, Dept. of Eng. Cybernetics, NTNU
Co-Supervisor: Artur Zolich, Dept. of Eng. Cybernetics, NTNU

Preface

The master thesis describes a development process that was made possible through the involvement of participants with different disciplinary backgrounds and experience. The supervisor for the thesis was Professor Tor Arne Johansen, and the co-supervisor was Artur Zolich. I would like to thank Tor Arne Johansen, who was the mastermind behind the project, both for his guidance and for inviting me to participate in this very inspiring project. I would also like to thank Artur Zolich for enlightening discussions and solid guidance based on his experience with development in the defense industry. This project would not have been possible without Peter Knutsen at the department's workshop. I would like to thank him for his positive attitude and thorough work in building the hardware cases for the system. Further, I would like to thank Frode Volden and Pedro De La Torre at Trondheim Biological Station for excellent assistance during testing of the vehicle. I would also like to thank AutoNaut for providing information and sharing ideas. I would like to thank my friend Amine Bouhouche for proofreading, fellow student Eivind Kjøsnes for guidance on solar technology and Sølve Sæter for great companionship and collaboration throughout the project. Last, but not least, I would like to thank my family for being supportive.

Bendik Olai Agdal,
Trondheim, June 2018

Abstract

The development, implementation, and testing of a control and monitoring system for a wave and solar-powered Unmanned Surface Vehicle (USV) are presented. In the future, the USV will be used for numerous science experiments and can be deployed for month-long unmanned operations in the ocean. A systems engineering approach focusing on stepwise development and testing was used to ensure high reliability. System safety and robustness was the primary focus during development. This was sought achieved by monitoring of internal system states and by implementing a fallback design. A polling loop with a switch case implemented state machine was programmed on a robust embedded computer to control the system. Several peripheral devices were interfaced to the main computer using different protocols. The USV was tested in the Trondheim Fjord, and the design behaved as intended. Some tuning remains to be done on fallback autopilot and the leak detection system, but early results indicate that the design and implementation are well suited for applications in USV systems where electric energy is limited, and a high level of robustness is required.

Sammendrag

Utvikling, implementasjon og testing av et kontroll- og monitoreringssystem for et ubemannet bølge- og soldrevet overflate-fartøy blir presentert. Fartøyet skal benyttes i en mengde forsøk og kan brukes i ubemannede opersjoner med flere måneders utstrekning. Systems Engineering utviklingsmetodikk ble benyttet i projektet. Stegvis utvikling og testing ble benyttet for å sikre høy pålitelighet. Systemsikkerhet og robusthet var primærfokusene under utviklingen. Dette ble søkt oppnådd ved monitorering av systemets interne tilstander og implementering av reserveløsninger. En tilstandsmaskin ble brukt for å implementere ønsket system-oppførsel. Det ble laget grensesnitt mellom hoveddatamaskinen og flere enheter ved bruk av ulike protokoller. Fartøyet ble sjøtestet i Trondheimsfjorden. Designet fungerte tilfredstillende. Noe fininstilling gjenstår på systemets autopilot og lekkasje-deteksjonssystem, men resultatene fra tidlig testing indikerer at systemdesign og implementasjon er velegnet for ubemannede overflate-fartøy hvor elektrisk energi er begrenset og høy grad av robusthet er påkrevd.

Contents

List of Figures

Nomenclature

ADCP	Acoustic Doppler Current Profiler
AIS	Automatic Identification System
AUV	Autonomous Underwater Vehicle
COG	Course Over Ground
FMEA	Failure Modes and Effect Analysis
FMECA	Failure Modes Effect and Criticality Analysis
MPPT	Maximum Power Point Tracking
PI	Proportional Integral
PV	Photovoltaic
PWM	Pulse Width Modulation
UAV	Unmanned Aerial Vehicle
USV	Unmanned Surface Vehicle
SHF	Super High Frequency
VHF	Very High Frequency

Chapter 1

Introduction

The Autonaut is a zero-emissions unmanned surface vehicle (USV) built for information gathering and distribution. This master project builds on the semester one report focusing on the same vehicle and extends into the high and low-level design, implementation, testing, integration, and verification of the system.

1.1 Motivation

The ocean-based economy is growing rapidly and is estimated to provide 40 million workplaces in 2030 (1). Securing sustainable use of the ocean while allowing for growth is regarded as one of the major challenges for the authorities in the coming years. Increased production in ocean industries must be balanced with the need for protection of the ocean (2). Sustainable exploitation of maritime resources relied on a good understanding of synergies and improved monitoring as well as greener technological solutions. If the efforts towards increased sustainability fail, today's and tomorrows ocean resources may be lost.

The AutoNaut project at NTNU will offer valuable new insight into technologies that are expected to allow for responsible economic growth. The five-meter USV is developed by AutoNaut Ltd in England and provides a green platform for ocean monitoring (3). Unlike conventional USVs, the AutoNaut relies on wave energy for propulsion. The unique mechanical fin system has zero emissions when it is operating and is not as limited by battery capacity as purely electric vehicles. Since the energy for vehicle propulsion is generated from waves, all the energy harvested from solar cells on deck can run the onboard electronics, allowing for month-long operations. Boats utilizing internal combustion engines typically require close monitoring and more frequent maintenance, making them less suitable for long-term unmanned operations. Such boats also have a considerably larger impact on the marine environment due to emissions.

It is not only the AutoNaut vehicle itself that is innovative: The payload carried on the vehicle is what truly allows for exploration of wide range of concepts and technologies. Several of the sensors can be used in conjunction to estimate physical and biological characteristics in the seawater such as algal blooms. This information can be sent to a decision-making unit, in real-time, that combines the information with that from other sources. For example, the USV is intended to be used in a system that seeks to avoid the harmful effects of dangerous algae by warning aqua farms before exposure. The USV may play one of the key roles in the algae warning system that is likely to also include small satellites, unmanned aerial vehicles (UAVs) and autonomous underwater vehicles (AUVs) (4).

Fish tracking is another use of the system. By capturing fish and fixing transponders before releasing them, the AutoNaut can be used as part of a fleet of USVs that can track the fish with onboard hydroacoustic receivers. This can be valuable for understanding fish behavior, for setting sustainable fishing limits and for more effective fishing.

Joint operations involving multiple USVs, AUVs and UAVs, are expected to play a larger role in the maritime sector in the future. The USV will carry a unit for underwater acoustic communication in addition to multiple radio communication systems. This will enable the USV to function as a communication node in linking underwater vehicles, surface vehicles and aerial vehicles in a network. (5)

Since the AutoNaut can be deployed for extended periods, without the need for frequent maintenance and refueling, it introduces a new approach to marine monitoring. Whereas buoys usually require extensive logistics for deployment and retrieval, the AutoNaut can easily be launched from a slipway and autonomously navigate to the area of interest. It has been shown that the AutoNaut can stay within a 35-meter radius from the desired position by circling (3). The ability to move to the desired position autonomously could also make the system feasible for monitoring of marine traffic and military surveillance.

In addition to the described use cases, the USV system will serve as a platform for further development of anti-collision systems and marine communication networks. Initially, the anti-collision system will be based on information from the maritime Automatic Identification System (AIS), but plans include the use of cameras. A master student will start working on a camera based object detection system in August 2018. Concerning communication, the USV will be built to enable integration of a wide range of systems: VHF-radio, Seatex MBR, and Ubiquiti M5 rocket S-band radios, satellite communication and mobile cellular network systems.

1.2 Scope

The focus of this master thesis is to develop an onboard system that handles essential functions such as power monitoring, power handling and error checking in addition handling fault situations. Because the USV is capable of month-long operations taking place far from the operators, the robustness of the system is essential. In the event of an error, failure or unexpected circumstances, there might not be anyone present to handle the problem.

Systems with similar external conditions are not uncommon, but unlike commercial systems, this system will be used for a number of science experiments. In such projects, thorough testing and validation before deployment might not be feasible and latent errors might result in failure in situations where the vehicle would be difficult to recover. To reduce the risks related to future projects using the system, a hierarchical structure is used for the onboard system. The focus in the master thesis is the overall high-level system design and the detailed design of the USV control subsystem.

1.3 Background

The master thesis builds on the project report (by the writer) from semester 1, which focused on understanding the user and system needs and proposed a design for the on-board control system architecture. In the project report from semester 1, the proposed design was briefly discussed in terms of strengths and weaknesses, but a theoretical framework for analysis was not used. Several high-level design changes were done early in the master project to mitigate some of the weaknesses that were pointed out. These changes in combination with the arrival of the USV and devices lead to re-thinking of many aspects of the low-level design.

Unless otherwise explicitly stated, everything that is presented and described was done as part of the master thesis, including all the figures and plots.

1.4 Outline

The report is organized in accordance with the system development approach. The purpose of the project is to develop a system that relies on technology associated with several different fields. Therefore the theory which makes up the grounds for decision-making has been included in the different chapters. This was done in order to improve coherence instead of attempting to cover all of it in one chapter solely dedicated to theory.

Chapter 2

Project Structure

This chapter describes the approach taken to ensure that the system will meet the expectations.

2.1 Systems Engineering

The primary motivation for using a systematic approach to system development is to control the cost, functionality, expandability, maintainability, and robustness of the final system. Systems engineering is a field devoted to bridging the gap between the user's operational needs and the engineered system. In systems engineering, the fundamental belief is that this is best achieved by providing a systematic framework for development. According to (6), attempts at performing big leaps from idea to solution often end in costly build-test-redesign loops.

Numerous development practices exist within the field of systems engineering. The key phases are usually the same, but variations seek to tailor a practice towards a specific field of application.

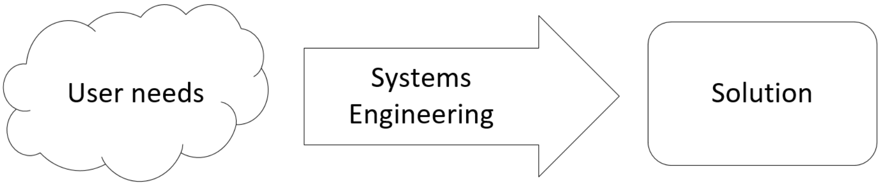


Figure 2.1: Systems Engineering

2.2 Standard for Best Practice

According to the D203 development standard, an integrated software dependent system is a system where overall performance is software dependant (7). At first glance, a maritime vehicle like the Autonaut might not seem to fit the definition well, but when assessing the characteristics of the system, the conclusion is clear. Firstly, the USV shall operate autonomously, which is only possible due to software. Secondly, unlike most vehicles, its only purpose is to gather information and distribute information, tasks that are also software centered.

The D203 standard is comprised of five steps: Basic Engineering, Engineering, Construction, Acceptance, and Operation. The different phases correlate with the fundamental systems engineering phases: Developing of concept and functional requirements, high and low-level design, implementation, integration, testing, verification and validation and operation. The D203 standard does not include decommissioning which is the final step in a system life cycle as seen from a systems engineering perspective.

Although the DNV GL-OS-D203 was helpful for developing user requirements before the master project, following the D203 development standard is very time-consuming since many steps should be meticulously followed. Since there are relatively few people involved in the project, it is possible to coordinate work and discuss solutions to ensure system integrity without going through all the procedures required by the D203 standard. Even though not all the procedures in the D203 standard were followed, the underlying principles of systems engineering, on which it builds, were used.

2.3 V-model for Development

The V-model for development captures the development process graphically. The level of detail increases downwards on the vertical axis, while project completeness increases leftwards on the horizontal axis. The model is said to be pseudo time-based (6). The point of time in which the activities are initiated correlates with their horizontal placement in the model. However, the integration and testing phase of the project may enforce changes in the project definition, making it necessary to redo parts of the activities associated with project definition and implementation. For example, if the current consumption of a device is found to be too high during integration and testing, the high-level design might require alterations. Therefore the model is not strictly time-based, but pseudo time-based.

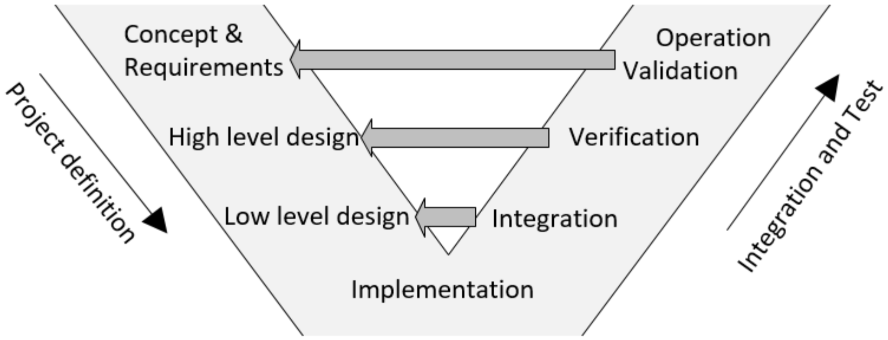
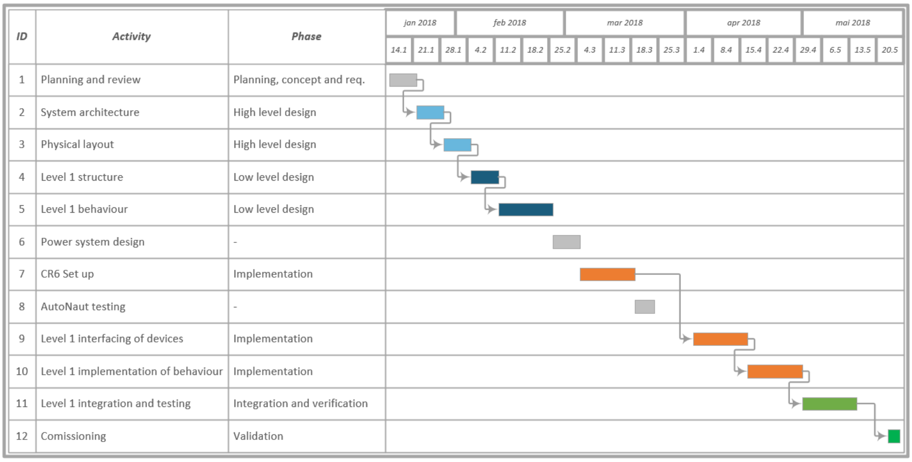


Figure 2.2: V-model for development

2.4 Project Activities

Throughout the project, the V-model for development was adhered to. The activities in the Gantt chart for the project can easily be traced to the phases that are fundamental in systems engineering and shown in the V-model for development.

The first activity in the Gantt chart consisted of planning for the semester and



ID	Activity	Phase
1	Planning and review	Planning, concept and req.
2	System architecture	High level design
3	Physical layout	High level design
4	Level 1 structure	Low level design
5	Level 1 behaviour	Low level design
6	Power system design	-
7	CR6 Set up	Implementation
8	AutoNaut testing	-
9	Level 1 interfacing of devices	Implementation
10	Level 1 implementation of behaviour	Implementation
11	Level 1 integration and testing	Integration and verification
12	Comissioning	Validation

Figure 2.3: Gantt chart for project

reviewing the concept and requirements as defined in the semester one project. This activity is associated with the concept development and requirements definition phase in the V-model. As the master project builds on the semester one project, only one week was needed for this.

Activity number two, system architecture, strictly belongs to the high-level design phase of the V-model. A complete proposal for system architecture was presented by the author in the semester one project, but due to drawbacks which were pointed out in the proposed architecture a re-evaluation was done as part of the master project. That led to a minor change in the high-level design.

After deciding on the system architecture, the process of planning the physical layout of the system started. The full process of designing the physical layout involves high and low-level design in the V-model. However, the detailed physical layout was carried out by the workshop at NTNU. The higher level physical design, on the other hand, is highly dependant on the system architecture and has implications for the further work. Therefore it was essential to coordinate and work together at the early stage.

After the physical layout of the main elements had been decided, the low-level structural design of the main elements in the overall system could be initiated. Since the high-level design was completed at this stage, focus was directed towards the level 1 subsystem, while the level 2 and level 3 subsystems were the focus for Sølve Sæter and Artur Zolich. Level 1 structural and behavioral design is part of the low-level design phase in the V-model.

The power system activity involved selecting appropriate devices for handling the PV panels, batteries and charging the system. The activity cannot be contained within one phase in the V-model. The selecting of components for the power system involved evaluation of requirements, high and low-level design and an assessment of implementation possibilities. Hence, it involved many of the activities related to the V-model, but on a smaller scale. Since the power management would be controlled by the level 1 subsystems computer, it made sense that it was done by in parallel to the level 1 low-level design. Since the design decisions relied on information from AutoNaut, the manufacturer of the PV-panels and several manufacturers of charge controllers, it was not possible to finish the activity at the same time as the low-level design for level 1.

Activity 7, 9 and 10 (orange) in the Gantt chart are all within the implementation phase. The first activity consisted of getting familiar with the operating system of the CR6 unit to understand how the desired behavior could be implemented. Activity 9 dealt with writing functions that interfaced the level 1 peripheral devices with the computer. During those stages, an embedded computer was also configured to simulate the level 2 subsystem so that the level 1 system could be tested. Activity 10 was the implementation of the desired level 1 behavior as defined in activity 5.

Note that activity 8 is not linked to the other activities. Activity 8 involved the delivery and acceptance test of the vehicle and included a course on how to best use the vehicle. Being able to discuss the planned system with the manufacturer and comparing it to the system they usually deliver was useful and led to some minor changes in the low-level design.

Activity 11 is related to the integration and verification phases of the V-model and consisted of testing the systems contained within level 1, but also included testing level 1 as an integrated part of the overall system and verifying that the system was performing as defined in the high-level design.

The final activity in the Gantt chart is commissioning and fits the validation phase of the V-model. At this point, the vehicle was tested in the Trondheim Fjord to validate the system requirements.

Figure 2.4: Arrival and commissioning, March 2018. Photo: Artur Zolich

Chapter 3

Concept and Requirements

The requirements for the subsystems that were developed in the project were based on work was done previously (8). The DNV-OS-D203 standard was then chosen as a foundation for the approach to develop system level requirements and to develop the concept of operation (8).

3.1 AutoNaut USV

The system is based on the 5-meter version of the AutoNaut USV. The USV will be able to obtain a speed of up to 4 to 5 knots, depending on the sea state. The internal volume is 750 litres, and it can carry a payload of 300 Kg. Three 100 W PV panels are fitted to provide electrical energy for the onboard electronics. The wave propulsion system is purely mechanical and uses fins attached to struts in the bow and at the stern of the USV (3). As the pitch angle of the USV changes due to waves, horizontal foils attached to the struts will generate a longitudinal force as water flows over the surface. The foils are attached on spring-loaded hinges that allow for the angle of attack to change according to the motion. There is no way of changing the dynamics of the wave propulsion system during operation. Therefore a forward force will always be exerted on the USV if there are waves. On the stern strut, there is a rudder and a thruster. The thruster is intended for very calm conditions, where waves cannot alone propel the USV, for example in harbors. It may also be used to increase the velocity of the USV in emergency situations.

AutoNaut Ltd. can fit the USV with a wide range of sensors and devices at request from the customer. NTNU ordered the 5-meter USV outfit with the following:

- CTD Sensor: Seabird CTD SBE49
- LED Backscattering Sensor: ECO Puck Triplet
- Oxygen Sensor: AADI O2 Optode 4835
- ADCP: Nortek 500
- Weather Mast: Airmar 120 WX
- Active Radar Reflector: Echomax Active-XS
- Navigation Light
- AIS (Receive and Transmit): Raymarine AIS650
- PV Panels: 3 x Solbian SP 104
- Batteries: 4 x Sonnenschein Gel Batteries (12 V, 63 Ah)
- Rudder control: DC-motor, encoder and motor controller with PWM control input
- Thruster: BlueRobotics T200 and motor controller with PWM control input
- 3 x Bilge pumps (12 V)

AutoNaut Ltd. normally delivers USVs that are outfitted with onboard electronics to control the system. However, since the intention at NTNU is to use the USV for new purposes, the USV was delivered without a control system so that a new one could be developed and fitted. The onboard electronics listed above were delivered with loose pigtails allowing for NTNU to fit connectors.

3.2 System Use Cases

The AutoNaut system is a multipurpose platform. The system use cases are shown in figure 3.1. Note that the actors in the diagram consist of both stakeholders and other (non-human) entities. On the left side of the diagram (3.1) the system stakeholders that will operate, utilize and perform development work on the system functions are shown. Students from the Department of Engineering Cybernetics at NTNU are currently developing anti-collision functionality for the USV bases on AIS, and future work on the subject include the use of computer vision (Pers. Comm.). In this regard, the USV has a role as a platform for developing new anti-collision methods which may also prove useful for other systems.

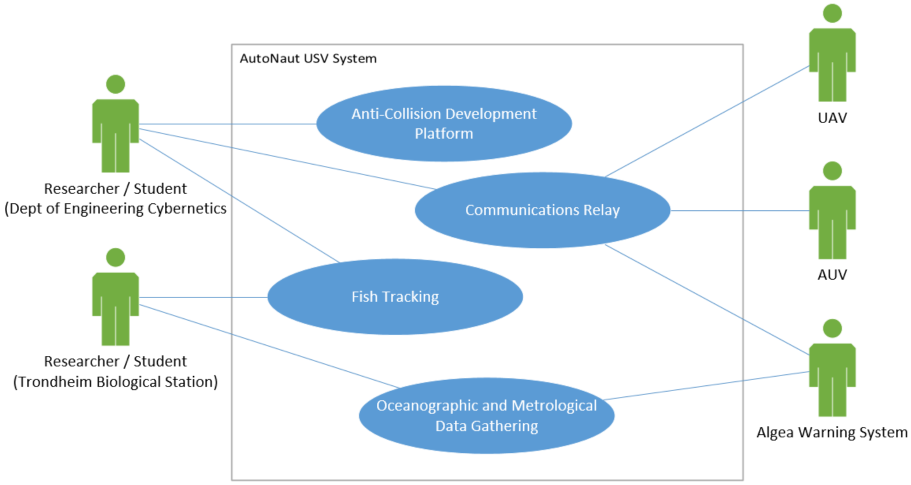


Figure 3.1: System Use Case Diagram

The USV will be outfitted with a sensor from Thelma Biotel that will allow it to track tagged fish (Pers. Comm.). Initially, the project will mostly involve engineers, but once completed the USV can be operated by Trondheim Biological Station with the aim of improving our understanding of fish behavior. Another important use case for the USV is to function as a communications relay. Students and researchers at the Department of Engineering Cybernetics will take on roles in developing the system. Since this development will lead to academic results with possible implications for other similar future systems, the use diagram emphasizes the connection between the stakeholders and the use case. On the right side of the diagram (3.1), other systems that use the USV system are shown. The motivation behind using the USV as a relay is rooted in the challenge of sending information from underwater to land and vice versa (9). In the future, the USV will be equipped with acoustic modems that allow for communication with AUVs. The USV will have several types of radio wave communication devices for air to air communication. This way data and control signals can be routed to and from subsea systems to systems such as AUVs to systems outside the area of operation. This will facilitate joint operations.

An algea warning system, presently in its early stages of development, will rely on several of the USVs use cases. Both tethering of aerial and subsea drones through the communication capabilities of the USV and also the data gathering capabilities of the USV could be important assets in the development of the warning system.

Several use cases have been described, but numerous possible application areas exist and are yet to explore. Better knowledge of the system behavior under varying circumstances should be gained before evaluating further possibilities.

3.3 System in Operation

The context diagram for the USV in operation (3.2) shows the entities that the USV will interact with during operation. Note that the diagram does not specify the communication interfaces in detail. The intention is only to provide stakeholders with a quick overview of the systems context and the passing of information between entities.

On the left side of the diagram (3.2) the operator station is shown. Information passes both ways: Control information is provided to the USV, and real-time system data is sent from the system. The purpose of the diagram is not to show how the information is passed from the USV to the operator station. In reality, it will not necessarily be passed directly. For instance, in some cases, it will go via a satellite system. Not including those details in the diagram is an example of how a system level context diagram can be simplified. Above the main block that represents the USV, a bi-directional communication link to a UAV is shown, indicating that the system will be able to pass and retrieve data from a UAV. It is made clear that GNSS will be used for position estimation. The USVs physical surroundings are represented by a block, and it is shown that the USV will obtain measurements of both meteorological and oceanographic parameters. Although a more advanced detection system is likely to be implemented in the future, the only mean of information passing between the USV and third-party ships and boats is via AIS. As indicated in the diagram, the AIS unit is enabled for transmitting and receiving. Lastly, a mechanism for fish tracking and communication with AUVs has been illustrated.

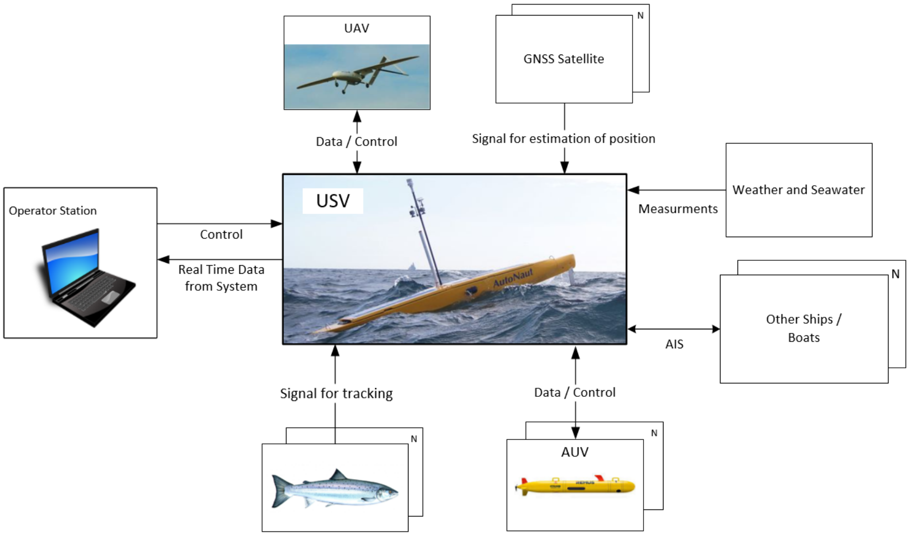


Figure 3.2: Context Diagram for USV in Operation

3.4 High Level System Requirements

High-level systems requirements were defined as part of the semester one report (8). As explained in chapter 1, several of the system functions are not prioritized yet and will be developed in the future. The high-level system requirements that were focused on in this project are the following:

- Manual Control Mode (A.REQ.2.3)
- Remote Insight in Real Time Data (A.REQ.2.6)
- Remote Control of Operatioinal Mode (A.REQ.2.7)
- Handling of errors, faults and failures (A.REQ.2.8)
- Onboard Power System (A.REQ.2.11)

In parallel to work presented in this report, Sølve Dahlin Sæter sought to meet the following requirements using LSTS toolchain:

- Waypoint Navigation (A.REQ.2.1)
- Automatic Collision Avoidance (A.REQ.4)

The requirements regarding oceanographic and meteorological data gathering capabilities (A.REQ.2.5) were taken into account in the high-level design (Chapter 4), but development and implementation of such features will be performed at a later point.

In the process of developing the system requirements (8), the requirements were allocated to proposed, and more detailed requirements were created under each high-level requirement. However, due to changes in the initial high-level design (8), as briefly described in Chapter 4, changes in the requirements to the different subsystems also had to make.

Chapter 4

High Level Design

The high-level design is based on previous work (8). However, alterations were made in order to mitigate a drawback in the earlier proposed design, namely that the oceanographic and meteorological sampling functions were distributed on two computers.

4.1 Layered Approach

The high-level design proposed in (8) was limited to two levels. A low-level system for handling power management and basic control function (hereafter referred to as level 1) and a higher level system running a Linux OS with Dune software for navigation and collision avoidance (hereafter referred to as level 2). Scientific data gathering functionality would be distributed between the two layers. Early in this project, it was decided that a third layer would be added to handle scientific measurements and calculations (hereafter referred to as level 3). The added system layer comes at the cost of increased power consumption but decreases the coupling in the design.

Layer	Requirements	Computer	OS / Program
Level 3	A.REQ.2.5 A.REQ.2.6	Technologic Systems TS-7970	To be decided
Level 2	A.REQ.2.1 A.REQ.2.4 A.REQ.2.6	BeagleBoneBlack (Industrial version)	Linux Glue OS / DUNE
Level 1	A.REQ.2.3 A.REQ.2.6 A.REQ.2.7 A.REQ.2.8 A.REQ.2.11	Campbell Scientific CR6	CR6 OS / Custom (PC400)

It would have been possible to implement the full range of functions on one or multiple embedded computers running a real-time operating system. Such a solution would have lead to lower power consumption for the onboard control system. In fact, that solution would have been more in line with the system AutoNaut Ltd. normally provide their customers (Pers. Comm).

The multi-layered approach decreases the coupling in the design. Particularly in a system that will see a lot of development of new functionality, that is a strong system characteristic. The goal is that anyone that seeks to implement new features to the USV system will only need to understand one system level. For example, if someone wants to test a new solution for environmental monitoring, that system can be implemented on the level 3 computer. The intention is that the level 3 computer will be available for scientific use, the level 2 computer will be available for navigation and collision avoidance functions while the level 1 performs strictly necessary system functions such as power monitoring and leak detection. The interface between levels is strictly defined so that an error or a failure at one level will not propagate through the system.

The other fundamental reason for developing a multi-layered system is to allow for graceful degradation in low energy situations. Since the only energy source for the onboard electronics is solar, situations, where energy must be conserved, might arise. In such cases level 3 can be turned off without losing safety-critical functions. In a worst-case scenario, level 2 can also be turned off.

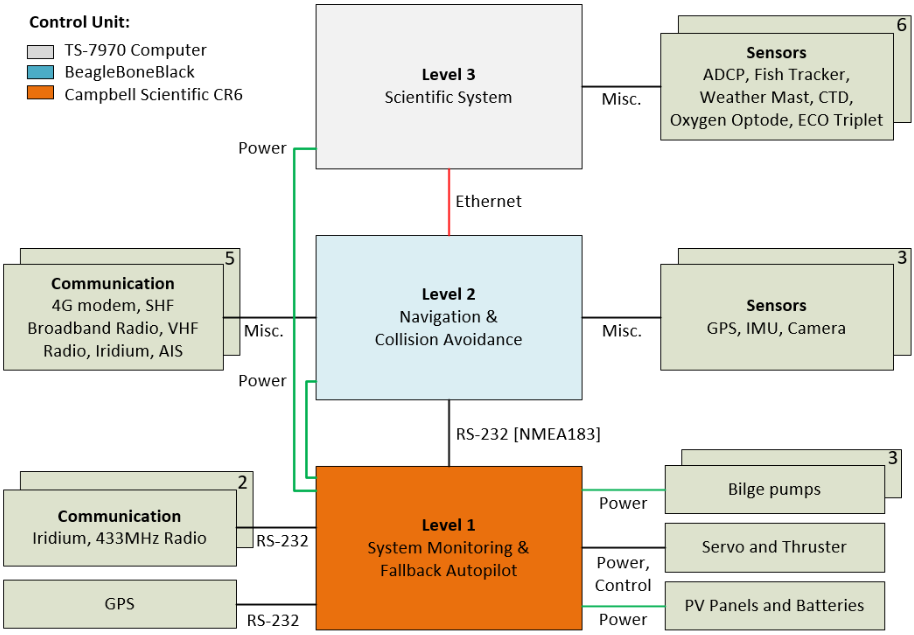


Figure 4.1: High Level Structure Diagram for USV

The high-level structure diagram (4.1) shows how the three system layers are connected to each other and the devices they interface. Note that the diagram is simplified and only shows the main system structure. For details about specific interfaces, the low-level hardware diagrams (appended) should be referred to.

Level 3 is intended for scientific use and does not hold functions that are important for navigation, collision avoidance or system monitoring. The system is powered by level 1, which means that a decision can be made to deactivate the level 3 system. This can either be performed automatically if energy must be conserved to maintain operation of safety-critical functions, or it can be done from the remote operating station. The Technologic Systems TS-7970 computer that level 3 is built around will be interfaced with the ADCP, fish tracker, weather mast, CTD, oxygen sensor and

ECO Triplet. The level 3 computer will be connected to level 2 via Ethernet. The level 3 computer will have access to communication devices via the Ethernet connection. Initially, a Ubiquity Rocket SHF Broadband radio will be used. The device creates a transparent link between the devices on the network (remote operator station, level 2 and level 3). Level 3 has had little focus on the master project, but level 2 has been prepared to interface with level 3 and developed is scheduled. Level 1 will not interface level 3 directly but will supply its power.

Level 2 is built around a BeagleBone Black (industrial version) and is dedicated for handling navigation and collision avoidance. As shown in Diagram 4.1, level 2 will interface several communication devices and different sensors. At the current stage of completion, level 2 has been interfaced with Broadband Radio, AIS, and GPS, but is expected to be outfitted with the remaining devices soon. Level 2 is powered from level 1. This allows the system to automatically shut down level 2 in case of detected errors or in case energy conservation is required. It also allows for a hard reset of the level 2 computer and devices. The functions can be performed from a remote operator station or automatically by the system in special cases. An adoption of NMEA1083 at RS-232 voltage levels is used for the data link between level 1 and level 2. The standard was chosen on the following basis: It has a low power consumption, the bandwidth requirements are low, it is human readable, and the signal can be tapped for easy debugging. Ethernet, SPI or I2C could also have been used with numerous possibilities for different high layer protocols. Level 1 interfaces the motor and servo control unit provided by AutoNaut, the three bilge pumps, and the PV panels and batteries. Additionally, level 1 interfaces a GPS for fallback purposes and two communication devices: Iridium and a 433 MHz short-range radio.

An example of an operational scenario can increase the understanding of the chosen high-level design. An algae bloom detection system could consist of a network of UAVs and USVs. Data from different sensors in the network could be transmitted to the level 3 scientific system in a USV for analysis. Based on the analysis, the onboard scientific system (level 3) could create new sets of waypoints that would be distributed. The USVs new set of waypoints would be transferred from the level 3 system to the level 2 system which handles navigation and collision avoidance. The level 2 system repeatedly calculates the desired rudder angle and thruster force and outputs to level 1, while level 1 handles actuator control and device monitoring to ensure system integrity.

4.2 Communication Links

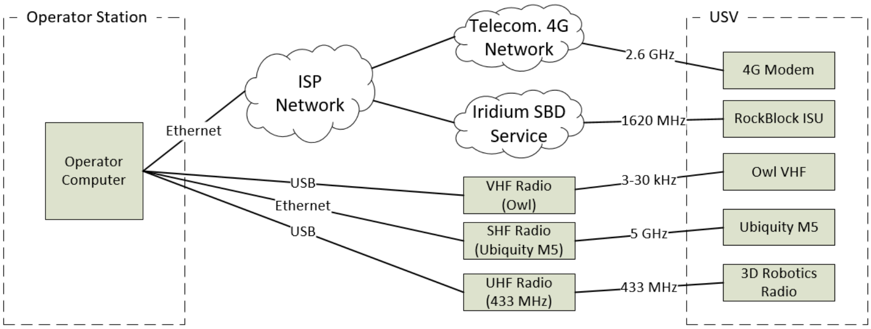


Figure 4.2: Communication Links between USV and Operator Station

There are several communication links between the USV and the remote operator station. Between level 1 and the remote operator station, there are two communication links: An Iridium system and a short-range radio system. The Iridium system is intended as a fallback solution in case of failure of the level 2 system. Level 2 and level 3 will be linked to the operator via a Ubiquity Rocket SHF Radio device initially. VHF radio, Iridium and 4G modem is also planned.

The short range radio system will be used for manual control of the USV during launch and retrieval. The 433 MHz transmitters are reliable within a few hundred meters. One unit will be connected to a PC via a USB port and function with a basic control software made by Artur Zolich. The other unit will interface with the CR6 computer in level 1 of the USV onboard system.

Level 1 will also interface with an Iridium transceiver. Iridium is the only satellite communication system provider with global coverage and is known for high-reliability (10). The selected Iridium transceiver is for short burst data (SBD) and will allow for reporting of parameters such as the state of the onboard systems and the USVs position, but also the ability for the remote operator to send simple commands to the onboard system.

Level 2 and level 3 will be linked to the remote operator via a super high frequency (SHF) broadband radio. Within the line of sight, the Rocket M5 SHF radio has a bandwidth of 150 Mbps (11). This will allow the remote operator to send and receive data and control instructions to the USV. The broadband connection will also be able to handle data streams from AUVs linked via the USV. However, that functionality is not yet developed.

A very high frequency (VHF) radio transponder will be able to receive and transmit data beyond line of sight, but with a lower data bandwidth than the SHF radio. The VHF radio has not yet been integrated into the system. The Level 2 system will also feature iridium communication, but it will only be used in areas where other options are unfeasible due to the cost.

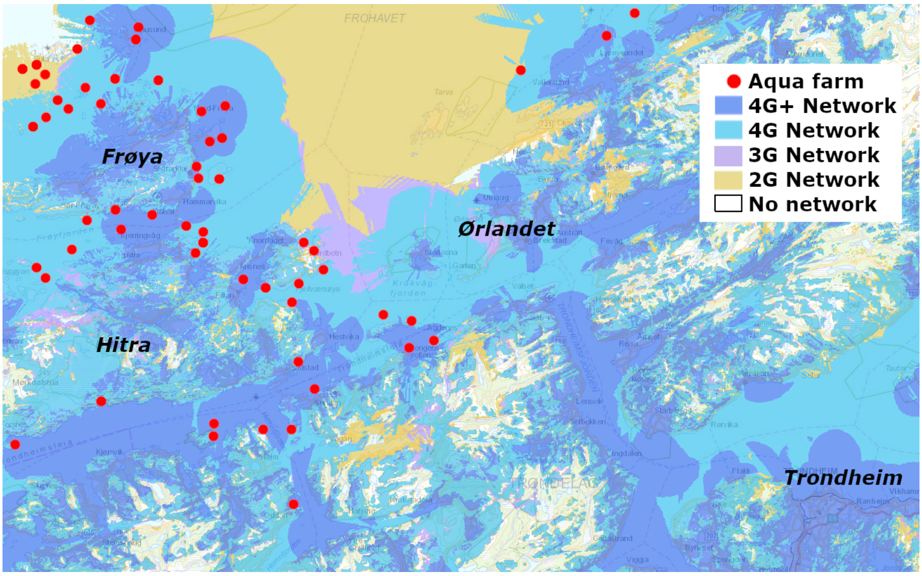


Figure 4.3: Telecom. provided reception and aqua culture facilities

In many areas, data service provided by telecommunication operators is available. Figure 4.3 shows a section of Telenor's map (12) of data reception in the coastal area near Trondheim. The red dots in the map (4.3) are aquaculture facilities for salmon, extracted from The Directorate of Fisheries' map for aquaculture along the Norwegian coast (13). In the fjords and sheltered waters, there is good coverage of 4G and 4G+. Most of the areas without reception are areas between mountains. However, the range of the telecommunication network is limited, and in open waters, the coverage is limited to a few kilometers off the coast. This can be seen in the top left corner of the map (4.3. There are three principal reasons for using the preexisting telecommunication network provided by Telenor or Telia. Firstly, the network standards automatically handle handovers between antennas and the infrastructure is already provided. Secondly, the data speeds are relatively high (40 Mbps uplink with 4G+ reception) (14), and thirdly the subscription costs are low.

Chapter 5

Robustness Analysis

System safety and system robustness are related concepts. In general, increased robustness will reduce the likelihood of hazardous situations (15). An analysis focusing on hazards will not be undertaken as part of this project. The focus of the project is rather on developing the functionality that can help users avoid hazardous situations. System safety could be a master project in itself. This chapter therefore focuses on the robustness of the system, the systems ability not to fail.

The Failure Modes and Effects Analysis (FMEA) was performed after the high-level design had been chosen and was used in the development of the low-level design. The process of developing the low-level design and behaviour was not completely stepwise. In reality, the final design was a result of an interactive process that involved the FMEA. However, what is presented in this report is the analysis of the system configuration that was chosen in the end.

5.1 Theory and Method

System safety was a field that developed alongside systems engineering after World War II. The first application of disciplined system safety approaches was under the Apollo space program (15). The intention of following a procedure for increasing system safety is not necessarily to make the system as robust or safe as possible at any cost. By identifying and making system risks visible, decisions can be made in order to reduce those risks to a minimum within the project constraints. If a formal assessment of the system is not performed, there is no way to verify that the design risks have been minimized and the design integrity would be more dependent on the experience of the engineers (16). Failure Modes and Effects Analysis (FMECA) is recognized as suitable method for achieving the required visibility into the system. This also holds for extremely complex systems when implemented rigorously (16).

FMECA can be performed to varying degrees of detail. Some projects formally require a part level FMECA. Apart level FMECA identifies failure modes for each individual electrical component such as a single capacitor. In cases where modules employ a high degree of redundancy, a part level FMECA is not necessary (16). In these cases a functional level FMECA is sufficient. In cases where many root part level failures causes lead to identical failure in a block, there is no value in identifying the role of the single part on root level (16).

The devices chosen for the USV system are commercial off the shelf. Therefore, there is limited value in performing a part level FMECA. In some cases the devices have built-in redundancy or failure mitigation functionality (for instance the Campbell Scientific CR6), or the root part level failure will lead to failure of the device. Therefore a part level analysis would provide little value even if it was possible. A functional level analysis is adequate.

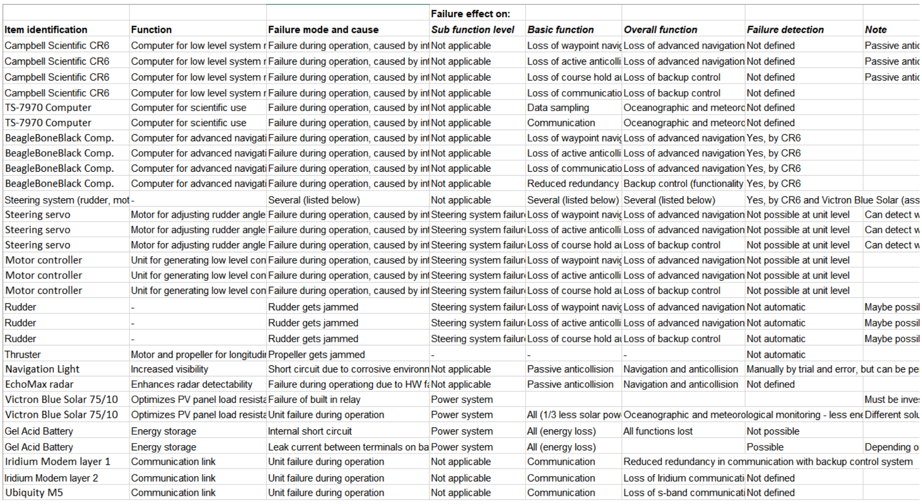
An FMECA procedure is carried out in two steps. Step one is Failure Mode and Effects Analysis (FMEA) and step two is the Criticality Analysis (CA) (17). For the USV system, only the first part of the analysis (FMEA) is carried out. The critically analysis requires documentation or estimations of Mean time between failures (MTBF) or probability of failure which is unavailable for several devices. A critically analysis that is made on the basis of unreliable data could result in decisions made on the wrong grounds.

The FMEA requires a clear system definition, which should include system diagrams and drawings, specifications, operational descriptions, environmental profiles and reports on reliability of system components and past or similar systems. Test results from equipment run under specific conditions is desired. If such results are unavailable, experience from use of the same or similar equipment in other projects or should be used (17).

After the required documentation has been obtained, reliability logic block diagrams of the system are constructed (17). The reliability block diagrams purpose is not to provide understanding of unit interconnections, but are intended to show the functional dependencies so that the effects of failures can be traced upwards to the top level of the system. The functional reliability block diagrams make up the foundation for the analysis together with the rest of the documentation. The diagrams are used to analyze the effects of each failure and its effect on the system.

5.2 Fault Table and Reliability Logic

Figure 5.1 shows an extract from the document that summarizes all the evaluated failures and their effect on the system functions. The most significant findings and conclusion from the FMEA are presented in this chapter.



			Failure effect on:				
Item identification	**Function**	**Failure mode and cause**	***Sub function level***	***Basic function***	***Overall function***	***Failure detection***	***Note***
Campbell Scientific CR6	Computer for low level system r	Failure during operation, caused by int	Not applicable	Loss of waypoint navig	Loss of advanced navigation	Not defined	Passive antic
Campbell Scientific CR6	Computer for low level system r	Failure during operation, caused by int	Not applicable	Loss of active anticolli	Loss of advanced navigation	Not defined	Passive antic
Campbell Scientific CR6	Computer for low level system r	Failure during operation, caused by int	Not applicable	Loss of course hold au	Loss of backup control	Not defined	Passive antic
Campbell Scientific CR6	Computer for low level system r	Failure during operation, caused by int	Not applicable	Loss of communicatio	Loss of backup control	Not defined	
TS-7970 Computer	Computer for scientific use	Failure during operation, caused by int	Not applicable	Data sampling	Oceanographic and meteoro	Not defined	
TS-7970 Computer	Computer for scientific use	Failure during operation, caused by int	Not applicable	Communication	Oceanographic and meteoro	Not defined	
BeagleBoneBlack Comp.	Computer for advanced navigati	Failure during operation, caused by int	Not applicable	Loss of waypoint navig	Loss of advanced navigation	Yes, by CR6	
BeagleBoneBlack Comp.	Computer for advanced navigati	Failure during operation, caused by int	Not applicable	Loss of active anticolli	Loss of advanced navigation	Yes, by CR6	
BeagleBoneBlack Comp.	Computer for advanced navigati	Failure during operation, caused by int	Not applicable	Loss of communicatio	Loss of advanced navigation	Yes, by CR6	
BeagleBoneBlack Comp.	Computer for advanced navigati	Failure during operation, caused by int	Not applicable	Reduced redundancy	Backup control (functionality	Yes, by CR6	
Steering system (rudder, mot	-	Several (listed below)	Not applicable	Several (listed below)	Several (listed below)	Yes, by CR6 and Victron Blue Solar (ass	
Steering servo	Motor for adjusting rudder angle	Failure during operation, caused by int	Steering system failur	Loss of waypoint navig	Loss of advanced navigation	Not possible at unit level	Can detect w
Steering servo	Motor for adjusting rudder angle	Failure during operation, caused by int	Steering system failur	Loss of active anticolli	Loss of advanced navigation	Not possible at unit level	Can detect w
Steering servo	Motor for adjusting rudder angle	Failure during operation, caused by int	Steering system failur	Loss of course hold au	Loss of backup control	Not possible at unit level	Can detect w
Motor controller	Unit for generating low level con	Failure during operation, caused by int	Steering system failur	Loss of waypoint navig	Loss of advanced navigation	Not possible at unit level	
Motor controller	Unit for generating low level con	Failure during operation, caused by int	Steering system failur	Loss of active anticolli	Loss of advanced navigation	Not possible at unit level	
Motor controller	Unit for generating low level con	Failure during operation, caused by int	Steering system failur	Loss of course hold au	Loss of backup control	Not possible at unit level	
Rudder	-	Rudder gets jammed	Steering system failur	Loss of waypoint navig	Loss of advanced navigation	Not automatic	Maybe possil
Rudder	-	Rudder gets jammed	Steering system failur	Loss of active anticolli	Loss of advanced navigation	Not automatic	Maybe possil
Rudder	-	Rudder gets jammed	Steering system failur	Loss of course hold au	Loss of backup control	Not automatic	Maybe possil
Thruster	Motor and propeller for longitudi	Propeller gets jammed	-	-	-	Not automatic	
Navigation Light	Increased visibility	Short circuit due to corrosive environn	Not applicable	Passive anticollision	Navigation and anticollision	Manually by trial and error, but can be pe	
EchoMax radar	Enhances radar detectability	Failure during operationg due to HW fa	Not applicable	Passive anticollision	Navigation and anticollision	Not defined	
Victron Blue Solar 75/10	Optimizes PV panel load resista	Failure of built in relay	Power system				Must be inve
Victron Blue Solar 75/10	Optimizes PV panel load resista	Unit failure during operation	Power system	All (1/3 less solar pow	Oceanographic and meteorological monitoring - less ene		Different solu
Gel Acid Battery	Energy storage	Internal short circuit	Power system	All (energy loss)	All functions lost	Not possible	
Gel Acid Battery	Energy storage	Leak current between terminals on ba	Power system	All (energy loss)		Possible	Depending o
Iridium Modem layer 1	Communication link	Unit failure during operation	Not applicable	Communication	Reduced redundancy in communication with backup control system		
Iridium Modem layer 2	Communication link	Unit failure during operation	Not applicable	Communication	Loss of Iridium communicati	Not defined	
Ubiquity M5	Communication link	Unit failure during operation	Not applicable	Communication	Loss of s-band communicati	Not defined	

Figure 5.1: Extract from the document that shows failure modes and effects.

The fault table as shown in 5.1 lists all the evaluated system items, their function, possible failure modes and cause and the effects on the system functions. As seen in the reliability block diagrams 5.2 and 5.3, the system's functionality has been organized in a hierarchy that consists of overall functions and basic functions. For the reliability block diagrams, three high-level overall functions were defined:

- Oceanographic and Meteorological Monitoring
- Advanced Navigation and Collision Avoidance
- Backup Control

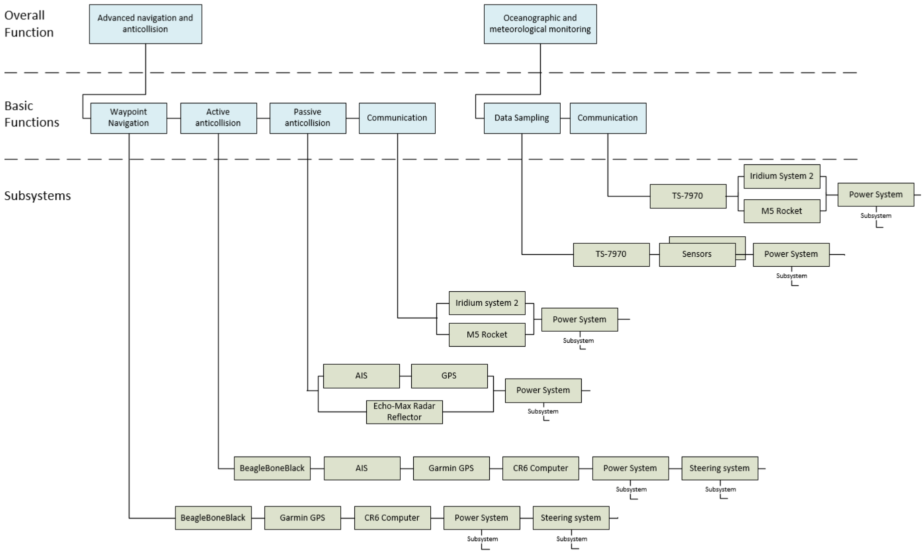


Figure 5.2: Reliability Block Diagram I

The overall functions are broken down into multiple essential functions. The basic functions require several subsystems to function. At this level it can be seen that several of the basic functions and thereby also overall functions, rely on the same subsystems, which dictates that failure of certain subsystems can lead to failure of multiple basic functions.

The overall functions align with the three structural levels of the onboard computer system. Oceanographic and meteorological monitoring will primarily be handled by level 3. Advanced navigation and anti-collision is handled by level 2 and backup control is handled by level 1. However, the systems are not fully separate or independent of each other. For instance the level 2 system handling the advanced navigation relies fully on multiple of the components associated with level 1. This is one of the reasons the reliability logic block diagrams are needed to perform an assessment of the overall robustness of the system.

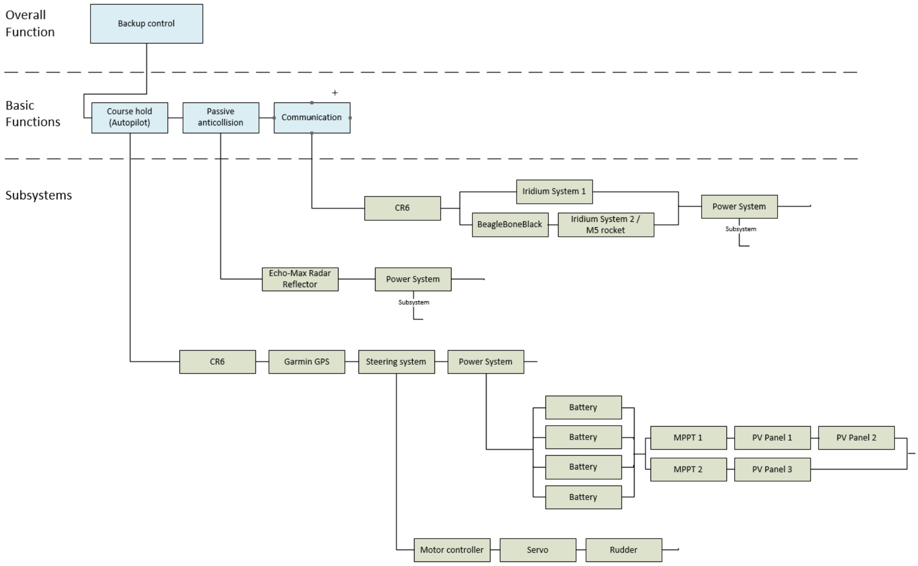


Figure 5.3: Reliability Block Diagram II

5.3 Failure Modes and Effects

In the following section, the identified failure modes as listed in the failure mode table will be discussed with respect to the basic and overall system functions. In addition, compromises and areas of marginal design will be discussed.

Since the power system affects all the basic functions and all the overall functions, it is the first topic in this analysis. In the diagram with backup control as the overall function, the power system has been expanded to show its components. The subsystem consists of four batteries in parallel. This indicates redundancy. If one battery fails, the subsystem will still be able to perform its function. Physically, the batteries are connected in parallel without any components between. Several failure modes are associated with ordinary acid-lead batteries, which is one of the reasons Absorbent Glass Mat (AGM) batteries were chosen for the vehicle. Opposed to acid-lead batteries, AGM batteries are not subject to internal short circuit due to mechanical failure inside the battery. Such a failure could have drained all the energy from the system if acid-lead batteries were used. AGM batteries are neither subject to leakage of explosive gas and internal soft short circuits due to acid stratification. In general AGM batteries are much more robust than ordinary acid-lead batteries (18). As can be seen in the failure mode table, leak currents between the terminals on the battery housings has been identified as a possible problem. If a layer of conductive particles accumulate on the battery housing, that could result in a small current between the terminals that would slowly drain the batteries. However, the batteries are placed inside the hull which is watertight, so it is very unlikely that salt particles or other conductive particles collect on the batteries unless there is a leak. In the event of a leak into the compartment containing the batteries, the batteries are likely to short-circuit long before discharging as the result of leak currents. Still, the batteries should be inspected after deployments. In the event of a significant leak current on the battery housing, it could be identified by the power management system.

In addition to batteries, the power system as shown in the reliability block diagram contains MPPT controllers and PV panels. If one MPPT controller stops functioning, the power from maximum two PV panels is lost, but the system will not fail. PV panels have several failure modes that could be detrimental to a system. If a PV panel is subject to partial shading, power will be dissipated in the shaded cells causing a temperature increase in those cells. Not only will the shaded cells reduce the current flow in the panel, the shaded cells can potentially melt and cause the panel to open circuit. To avoid these problems, bypass diodes are typically placed in parallel with cell arrays so that the shaded cells are bypassed instead of heated by the energy harvested in irradiated cells. According to the datasheet for the SP-104 panels, they have two built-in bypass diodes, so it is not necessary to take any measures to avoid this failure mode. Another failure mode occurs if the PV panel is shaded and energy flows from the battery into the PV panel. However, according to the datasheet for the MPPT controller, it has built-in reverse current protection that will avoid that from occurring.

The biggest threat to the power system is not the failure of components, but failing to estimate the expenditure of energy and the remaining energy. Especially in cases where little energy is harvested. If the remote operator does not make the correct decisions regarding energy use, the vehicle may end up in a state where the remaining energy is too low to run even the most important onboard functions. Therefore it is important to ensure that the operator is provided with sufficient data for decisions. For instance, the remote operator should be provided information about energy harvested compared to energy used along with an estimate of remaining energy on the batteries.

The second subsystem for analysis is the steering system. The steering system is necessary for both the advanced navigation and anti-collision and for the backup control overall functions. The entire steering system, as it is defined in the reliability block diagram, is delivered by AutoNaut. It consists of a motor controller, a servo and the rudder with adjacent mechanical components. The motor controller is delivered by RobotEq and the motor unit is custom for AutoNaut. Since the set up consists of software and hardware and has been made by AutoNaut, it is not possible to identify part level failure modes. However, AutoNaut has tested the system at their facilities for 90 days continuously alternating the rudder deflection between the maximum extents. Since AutoNaut delivered the first USV in 2014, there have been no reports of failure in the steering system (Pers.Comm.). Since motor controller is placed in an IP68 rated box inside the watertight hull, it is seen as very unlikely that failures in the motor controller due to water should occur.

The overall function oceanographic and meteorological monitoring consists of two basic functions: Data sampling and communication. The data sampling function is composed of three elements in series, meaning that they must all function in order for the basic function to work. The first element is the power system which has already been discussed. The second element is a set of sensors. Since the meteorological and oceanographic monitoring function of the vehicle will consist of many sensors that might be active at different times for different missions, a simplification was made in the reliability block diagram. For specific monitoring tasks, a more precise logic block diagram could be made, showing exactly which sensors must function in order to run the system. In some cases, such as temperature monitoring of the ocean, there is a degree of redundancy since more than one sensor measures that parameter. However, due to the vast number of monitoring use cases, the reliability block diagram was simplified.

The sensor block, under the data sampling function, uses a set symbol, indicating that there is a set of sensors, but how the sensors rely on each other, whether they should appear in parallel or in series in the diagram, has not been defined. More specific reliability block diagrams can be made for data sampling prior to future deployments with specific intents for environmental monitoring. The final block in series with the sensor block and the power system block, is the TS-7979 block. This block symbolizes the onboard scientific data processing computer. The sensors will be interfaced to this computer, so if it fails, the data sampling capability of the oceanographic and meteorological monitoring system will fail. One of the selection criteria for the TS-7970 computer was robustness. The computer has industrial temperature rating and will be protected as it is mounted inside an IP67 rated box in the watertight hull.

The second basic function that must work in order for the overall function oceanographic and meteorological monitoring function to remain intact, is communication. As can be seen in 5.2, the communication function is held by four blocks. The power system is required, the TS-7970 science computer is required and either iridium system 2 or the M5 rocket is required to function. The two former units have already been discussed. When it comes to the communication system, it is important to note that there will be areas where the M5 rocket based communication will be out of range. In this case, the communication system redundancy for the oceanographic and meteorological monitoring function is lost.

The Iridium system and the M5 rocket system that is used for the communication channel between the remote operator and the vehicle, are acquired as commercial of the shelf components and will not undergo any modifications other than software configuration. Therefore a more in-depth analysis of the units will not be performed.

Advanced navigation and anti-collision requires the most complex array of subsystems for adequate performance. The communication sub-function requires the power system to function in addition to either iridium system 2 or the M5 rocket communication system. As mentioned, redundancy in the communication system is lost if the M5 rocket system is outside range.

The anti-collision functionality is divided into two separate basic functions: Active anti-collision and passive anti-collision. Passive anti-collision deals with anti-collision in relation to other vehicles and rely on the other vehicle to take action. In its essence, the passive function that leads to anti-collision is detectability. Due to the small size of USV, it relies on an active radar reflector for creating a sufficient radar signature for other boats and ships. The active radar reflector requires the power system to function. The other element in the passive anti-collision system is the AIS. The AIS device requires inputs from its GPS unit and the power system to function. As can be seen in the reliability block diagram, the USV loses both systems if the power system fails. The bright yellow color of the hull is also a measure to increase detectability, but under rough conditions with waves and seaspray or during night time it will not have effect.

Waypoint navigation requires a series of five units to function. Two of the units are the power system and steering system which has already been discussed in detail. Further, the waypoint navigation function requires the BeagleBone Black computer, the Garmin GPS unit and the Campbell Scientific CR6 computer. The BeagleBone Black computer is of industrial grade to ensure sufficient robustness in regard to temperature and will be mounted inside an IP67 rated case inside the watertight hull. The GPS unit will be mounted outside the watertight box, but has IPX7 rating. Even though it is watertight, it is less protected than the other units, so after extended deployments with rough conditions the unit should be inspected for any signs of weakness. All the components are commercial off the shelf, so a part level analysis is not possible or necessary.

The active anti-collision is the function that requires the most extensive array of units to function. It requires all the units and systems that are required for waypoint navigation in addition to the AIS. The AIS consists of a unit that will be mounted inside a watertight IP67 box and an external antenna mounted to the mast.

The backup control system function is based on the same components as the advanced navigation and anti-collision system. There is no redundancy on hardware level. The components are the same. The difference is that the BeagleBone Black computer is not part of the subsystems and the communication system is different. The communication sub-function requires the Campbell Scientific CR6 computer to be operational, the power system and either iridium system 1 or the Beaglebone Black computer and the M5 rocket system. This means that the backup control system can still be accessed in case of failure of the Iridium System as long as the BeagleBone Black computer is operational with the SHF radio. It might seem unnecessary as the only clear difference between the other basic functions contained in the backup overall function versus the advanced functions is the BeagleBone computer. However, level 1, which handles backup control, also provides other functions such as power monitoring which has not been included in the reliability analysis. Therefore it could be useful to communicate with the system even when the BeagleBone is working.

The course hold function in backup control requires a very similar set of devices compared to the waypoint navigation function in advanced navigation. The only difference is that the BeagleBone Black is not required. Since the software is different it is not unlikely that the backup hold still works in cases where advanced navigation does not.

5.4 Robustness Discussion

The FMEA shows how failure of devices will lead to failure of system functions. The power system is necessary for every onboard function. However, it is not regarded as an area of marginal design because there are no single points of failure in the power system. However, it was noted that the biggest overall system threat related to the power system, was over expenditure of power. The remote operator should therefore be provided with data regarding the power state of the PV panels, the load power and the remaining energy on the batteries. That way the remote operator can make decisions regarding power usage and avoid reaching a critical low point on remaining energy forcing a system shut down.

The steering system is regarded as a potential single point of failure for navigation and active collision avoidance functions as there is only one rudder and one steering servo. However, since the system has been tested rigorously, it is not regarded as an area of marginal design.

The CR6 computer is a very vital system component. If it fails, all navigation functions are lost. The unit itself was selected for its proven reliability and use in systems across the globe, but software faults still pose a threat. Therefore, precautions must be taken when developing the software and it should be tested rigorously. One way of mitigating this area of marginal design in the USV system, would be to connect the BeagleBone Black computer to the control signal wires that go to the motor controller and rudder servo. By doing so it would be possible to bypass level 1 and control the thruster and rudder servo directly from level 2 in the event of failure of the CR6. The benefit of that solution is clear, but there are two drawbacks which constitute the reason for designing the system as it is. Firstly the increased hardware and software complexity increases the risk of faults, secondly the system cost and development time would increase.

The risk of software errors in the advanced navigation system is the reason for having a fallback control system. Since the advanced navigation system that runs on the BeagleBone Black computer is subject to further development by different people in the time to come, it is not seen as unlikely that faults may arise. The advanced navigation and collision avoidance system is also inherently more exposed to mistakes leading to failure due to its complexity. Therefore the fallback control system implemented on level 1 must be able to take over control in the case of such events.

Chapter 6

Detailed Design

This chapter covers the detailed design of the system's structure and behavior. The detailed design of level 2 and level 3 is not covered in this report because it was not within the scope for the detailed design for the project.

6.1 Level 1 Requirements

The DNV GL-D203 standard describes a best practice for developing integrated software dependent systems. The standard was used in the project (see chapter 2.2). In accordance with the D203 standard (7), detailed requirements were made for the level 1 system based on the overall system requirements. Although the high-level system requirements are the same as in (8), the detailed requirements differ from the original proposal because the higher level design was altered in the initial phase of the project. Findings in the robustness analysis (Chapter 5) also led to additional functional requirements to the low-level design.

System Requirement Description	ID	Subsystem Requirements	ID
Onboard Power	A.REQ.2.11	12 ± 2 V Output	B.REQ.2.11.1
	A.REQ.2.11	Load power monitoring	B.REQ.2.11.2
	A.REQ.2.11	PV panel power monitoring	B.REQ.2.11.3
	A.REQ.2.11	Remaining Energy Estimation	B.REQ.2.11.4
	A.REQ.2.11	Disabling of device power	B.REQ.2.11.5
	A.REQ.2.11	In-port charging	B.REQ.2.11.6
Error handling etc.	A.REQ.2.08	Device Error Monitoring	B.REQ.2.08.1
	A.REQ.2.08	Level 2 Failure Monitoring	B.REQ.2.08.2
	A.REQ.2.08	Leak detection	B.REQ.2.08.3
	A.REQ.2.08	Bilge pumps control	B.REQ.2.08.4
Control of Op. Mode	A.REQ.2.07	Remote Control Interface Protocol	B.REQ.2.07.1
	A.REQ.2.07	Manual Control Mode	B.REQ.2.07.2
	A.REQ.2.07	Level 2 Control Mode	B.REQ.2.07.3
	A.REQ.2.07	Fallback Autopilot Mode	B.REQ.2.07.4
Manual Control	A.REQ.2.03	Remote Control Interface Protocol	B.REQ.2.03.1
	A.REQ.2.03	Rudder Angle Control	B.REQ.2.03.2
	A.REQ.2.03	Thruster Control	B.REQ.2.03.3
	A.REQ.2.03	Disabling of Power for Devices	B.REQ.2.03.4
Remote Data	A.REQ.2.06	Iridium Communication Link	B.REQ.2.06.1
	A.REQ.2.06	Radio Communication Link	B.REQ.2.06.2
	A.REQ.2.06	Output System Energy Parameters	B.REQ.2.06.3
	A.REQ.2.06	Output Position, COG and SOG	B.REQ.2.06.4
	A.REQ.2.06	Output Leak and Error Status	B.REQ.2.06.5

6.2 Power System

The USV was delivered with four 12 V 63 Ah batteries and three Solbian 104 W solar panels. The power system had to be designed as part of the project. Voltage requirements for all the onboard devices where identified. The majority of devices, including the high energy ones, requires around 12 V. Therefore it was decided to wire the batteries in parallel, providing 12 V.

There are numerous possibilities when designing a power system. The most important choices are Wiring of the photovoltaic (PV) panels, selection of charge controller and selecting battery voltage. AutoNaut Ltd. normally delivers their USVs with three Genasun GV-10 MPPT controllers connected independently to the PV panels (Pers. Comm.). Such a configuration provides good redundancy, and the Genasun controllers are robust and relatively low cost. It was found that a different solution was better suited for this project.

The first step that was taken when designing the power system was to assess the supplied PV panels. The PV panels on the USV are Solbian SP104. Their maximum output power rating is 104 W. Maximum open circuit voltage is 21.8 V, and Maximum short circuit current is 6 A. The panels consist of four cell arrays and has two built-in bypass diodes. The output power of a given PV panel is dependant on multiple parameters. The most significant parameters are solar irradiation and observed impedance, but cell temperature also makes a significant impact on power output (19). A charge controller configuration should be selected depending on the expected output power.

There are two main types of charge controllers. PWM (Pulse Width Modulation) controllers and MPPT (Maximum Power Point Tracking) controllers. When using PWM controllers, it is advisable to wire the set up so that the PV panel voltage is close to the voltage of the battery system. If the panel voltage becomes higher than the charge voltage of the batteries, the PWM controller will start switching the connection between the PV panels and batteries on and off rapidly to avoid overloading the batteries. MPPT controllers, however, have built-in inverters and can step the voltage up or down before supplying the batteries. This is beneficial because the PV panel output varies with the observed load impedance. Since an MPPT controller has a built-in inverter, it can change the impedance that the PV panel observes according to the maximum power point (19). Simply put, MPPT controllers will always achieve equal or higher power outputs than PWM controllers but are more expensive. Since the USV will operate in conditions that vary in terms of irradiance and temperature, the added cost of choosing MPPT controllers will pay off in terms of increased yield.

There are three fundamentally different ways of wiring the power system: Parallel wiring, series wiring, and independent wiring. Parallel wiring results in a lower input voltage to the controller and higher current. The input current may reach 18 A ($3 * 6$ A short circuit current). The larger the current, the larger the energy loss on the wires. If a PWM controller were to be used, this set up could be beneficial as high input voltages would be avoided. With an MPPT controller, however, it will not matter as long as it is within the ratings for the controller. Due to the loss of the wires, however, it is not an advisable solution.

Series wiring results in a potential output voltage of 65.4 V. The maximum open circuit current would be 6 A. Since the PV panels have built-in bypass diodes, partial shading will not cause damage to the shaded panels and the reduction in harvested energy will not be severely limited by the shaded array. Having the three panels connected in series is regarded as a better solution than three panels in parallel for the particular application.

The final option is to connect the PV panels to different MPPT controllers. One benefit of that is increased system robustness as discussed in Chapter 5. Additionally, the configuration might lead to higher system yields. The drawbacks are added cost and increased self-consumption of the power system.

Some rough estimates of the yields under standard conditions for the different configurations were performed based on the datasheet for the PV panels and test results by the panel manufacturer on very similar panels. The test was performed on the Solbian SP100 panels that are very similar to the SP104 panels. In the test, two panels were connected in series under standard test conditions to measure the effect of partial shading (20). The test was performed by the manufacturer using a sun simulator for consistency. Each panel consists of two cell arrays, each with a bypass diode. Hence, the tested configuration consisted of four cell arrays in series with the

MPPT controller. With all four arrays irradiated, $V_{max} = 36.81$ V, $I_{max} = 5.58$ A and $P_{mp} = 205W$. That means that one cell array contributed 9.20 V (each cell 0.575 V). When two cell arrays were shade affected, resulting in activation of the bypass diodes, $V_{max} = 15.43$ V and $P_{mp} = 90W$. If each panel had a separate MPPT controller, the expected yield of the irradiated subsystem would be $P_{mp} = 103W$ ($V_{max} = 18.4$ V and $I_{max} = 5.58$). The yield of the shade affected subsystem would be slightly negative due to self-consumption of the charge controller. A simplified model of the two alternative subsystems is shown in figure 6.1. The inactive panels are grey, and the irradiated panels are white. Note that the shaded panel in series with the irradiated panel is bypassed with a diode. The loss in the system where two panels are connected in series is due to the forward voltage drop over the activated bypass diodes (20).

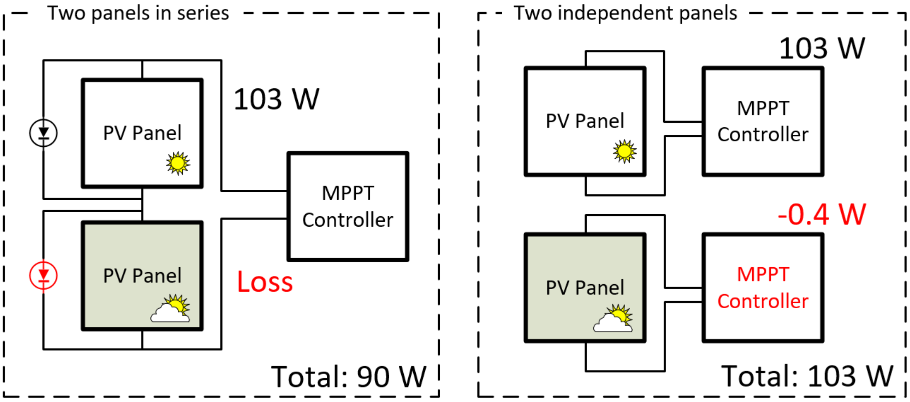


Figure 6.1: Partial shading on PV panel system.

The solution would seem obvious for a system where high energy yield is more important than low cost. However, several other factors regarding the power system configuration must be regarded. An important question is whether or not the bypass diodes will be activated due to the shadows. Partial shading effects could not be tested prior to ordering charge controllers because of project time constraints, but according to the technical team at Solbian (Pers. Comm.) deep shadowing on any out of the 16 cells in an array will result in activation of a bypass diode. Based on that information, it was assumed that shadows cast by the mast of the vehicle could result in activation of bypass diodes depending on solar irradiation strength and sun angle.

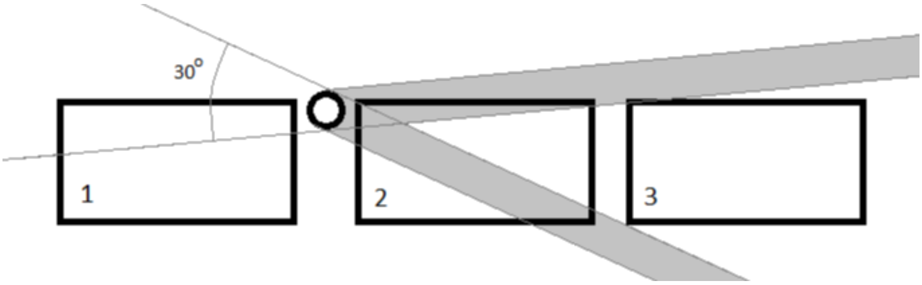


Figure 6.2: Mast shadow on PV panels.

Based on drawings of the USV, it was found that it is unlikely that the PV panel fitted near the bow of the USV will be partially shaded by the mast. Figure 6.2 shows that panel 3 will be fully irradiated unless the sun is within a 30-degree sector aft of the vehicle. The two panels adjacent to the mast may receive shading more often.

An issue that must be considered regarding partial shading and activation of bypass diodes is the resulting output voltage from the panels. Each Solbian SP104 PV panel consists of two arrays with separate bypass diodes. Under standard test conditions, the optimal output from one panel is 18.2 V. In the event of activation of one bypass diode, the output voltage will be 9.2 V, which is lower than the battery voltage. This means that a step-down charge controller will be unable to utilize the PV panels output. Many step up controllers, such as the GV-Boost 12, can handle an input voltage of 9.2 V but would be unable to handle an input above 12 V. Although they exist, buck-boost charge controllers are uncommon.

Diagram 6.3 shows the power system configuration with selected components. The selected power system delivers redundancy, mitigates the detrimental effects of partial shading and allows for power monitoring. Two step down MPPT controllers is used in the power system. Panel 3, which is far from the mast, will be connected to one controller because it is unlikely that the internal bypass diodes are activated due to shading, meaning that the panel output always will be higher than the required

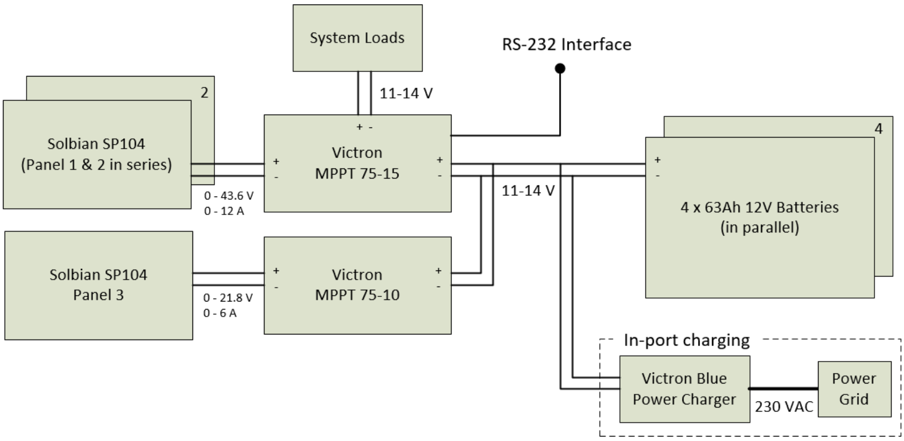


Figure 6.3: Structure Diagram for Power System

input voltage for the controller. The panels near the mast, which might be subject to partial shading, are connected in series to another step-down MPPT controller. That way, the charger's input will always be higher than the minimum voltage requirement, even if both arrays in one panel are bypassed. A Victron Blue Solar MPPT 75-15 was chosen for the single panel and for the two panels in series.

The Victron Blue Solar MPPT 75-15 is able to detect global maximum power points. Detection algorithms used in some MPPT controllers only detect local maximum power points and are therefore less suited for applications where shading can occur. In systems that are not subject to partial shading, it does not matter. Further, the Blue Solar controller is designed to function in a network of devices in their product range. Since Victron provides the application programming interface (API) for their devices, the Blue Solar can be integrated into the USV system. Most MPPT controllers only have LEDs to indicate internal status and cannot be interfaced with a computer. The Blue Solar MPPT 75-15 has a load output where current and voltage is measured. It also measures the battery voltage and the voltage and current on the panels. Since only one of the controllers will be interfaced with the computer, the power generated from panel 3 must be estimated based on the input from panel 1 and 2. Multiplying the measured power from panel 1 and 2 with a factor of 3/2 is a safe estimate because it is very unlikely that panel 3 will be affected by partial shading like the panels near the mast.

6.3 Structural Design

The structural design of level 1 was done in collaboration with Artur Zolich and Peter Knutsen. Initial proposals were made as part of the master project, but the detailed structural design was made by Artur Zolich, and the physical arrangement and installation of components inside the chosen box were done by Peter Knutsen. That allowed for directing of attention towards behavioral design for the master project and made it possible to stay on schedule.

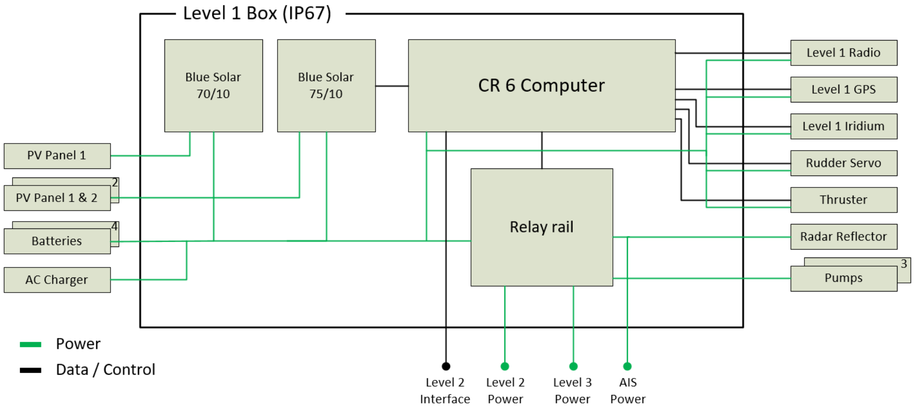


Figure 6.4: Level 1 Structure Diagram

Figure 6.4 provides an overview of the structural design for level 1. Note that some level 1 devices are powered via relays controlled by the CR6, whilst other devices are powered directly from the batteries. One could think that it would be beneficial to control all the device power with the use of relays. However, that may not be the case. Certain devices, such as the rudder servo and the CR6 are seen as so instrumental to systems operation (see Chapter 5) that they were connected to the battery without the option of turning them off with a relay. However, fuses protect the system from short circuits. The level 1 GPS and Iridium devices are not possible to turn off with relays but have sleep input pins that are connected to the CR6. In sleep mode, the devices retain their volatile memory to allow for low boot time. The Iridium modem also contains a capacitor that must be charged prior to transmitting messages. Once charged, the charge can be maintained with a minimal amount of power. These function would be lost if relays where used instead of the sleep pins. The decision not to use relays on all components is also a result of a limited number of input/output pins on the CR6 computer. Priorities had to be made.

6.4 Behavioural Design

Although the focus for the behavioral design is level 1, the subsystem must be seen as part of the overall system. The behavioral design focuses on how the interconnected devices in the subsystem should behave in order to meet the system requirements.

During development of the system, it was sought to keep complexity as low as possible while still meeting the system requirements. That was seen as important because of the system operators. Human operators tend to create simplified mental models of the systems they are responsible for (15). Discrepancies between the system and the operators mental model of the system can cause serious incidents.

A level 1 requirement that is related to interaction on the system level 2 is that level 1 should monitor for failure of level 2. As explained in Chapter 4, the level 2 computer will send a control signal to the level 1 computer. This will happen every second. If level 1 does not receive a verified control signal from the level 2 computer for a set amount of time, level 1 will assume that level 2 has failed. In the initial configuration, level 1 will not take any action apart from warning the operator and assuming control of the USV. Before more experience is gained, it is not sought to provide the USV with too much autonomy. In the future, however, a procedure for automatically rebooting level 2 in the event of suspected error can be added. At this point level 2 can be rebooted remotely, but the reboot must be initialized by the operator.

Figure 6.5 shows the fallback behavior that is designed to take place in the event of level 2 failure. The sequence diagram shows the communication between four entities: Operator, control station, level 2 and level 1. The operator can set waypoints using the computer at the control station. The desired waypoints are transmitted to the navigation software in level 2 by one of the system communication links (Chapter 4.2). In the USV the Level 2 software will calculate the desired rudder angle based on the waypoints and system information. Once every second (1 Hz) Level 1 will send system information to level 2 and expect an answer containing desired rudder angle within a set time. If an error or failure in level 2 stops the computer to respond to the requests from level 1, the level 1 system will assume that level 2 has failed and will enter fallback mode and take control of the USV. The timeout period for entering fallback can be adjusted. In initial trials, it will be set to 5 seconds. Once level 1 enters fallback mode, it will send a warning message to the operator using the Iridium modem. The fallback mode will be discussed in more detail.

Figure 6.6 shows the different operational modes that the level 1 subsystem can be in and the transitions between the states. When in the normal state (state 1), level 1 will control the rudder and thruster in accordance with the control messages received from level 2. In the fallback state (state 2), level 1 will assume control of the USVs rudder and thruster without being reliant on level 2. In manual mode (state 3), the remote operator will be in direct control of the rudder angle and thruster force. There is a total of six possible state transitions. The transition from NORMAL or FALLBACK to MANUAL will only take place if the operator sends an instruction to the system. If the connection is lost between the remote operator and the USV when in MANUAL, the USV will enter fallback mode. Fallback mode will be entered automatically if level 1 does not receive instructions from level 2. Note that a warning will be sent to the operator in the event of this transition. The transition from FALLBACK to NORMAL is also automatic and will occur as soon as level 1 receives a valid instruction from level 2.

In addition to the behavior that is already described, the level 1 system has to monitor the battery voltage, solar cell power, and load power, obtain GPS data and check for leaks. For leak detection, the onboard bilge pumps are used. Since the motors are inductive loads, the current will change based on the motor's resistance. At a

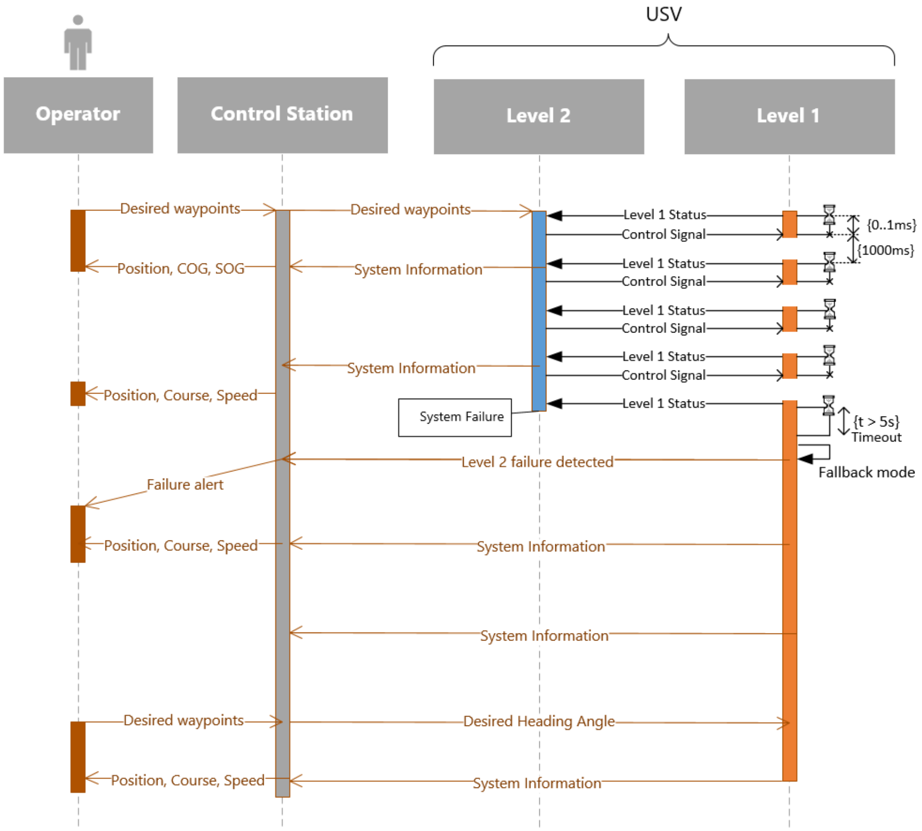


Figure 6.5: Sequence Diagram for System Fallback.

defined interval, the pumps will be activated. Based on the increase in current, the system will detect if water is being pumped.

When the system enters fallback mode, there are three different operating modes that the user can select from. Fallback mode 0 sets the rudder angle to zero and thruster to 0. Fallback mode 1 sets the rudder angle to 45 and the thruster to 0. Fallback mode 2 activates an autopilot that keeps the USV on a defined course angle.

The operator should select the desired fallback mode based on the circumstances.

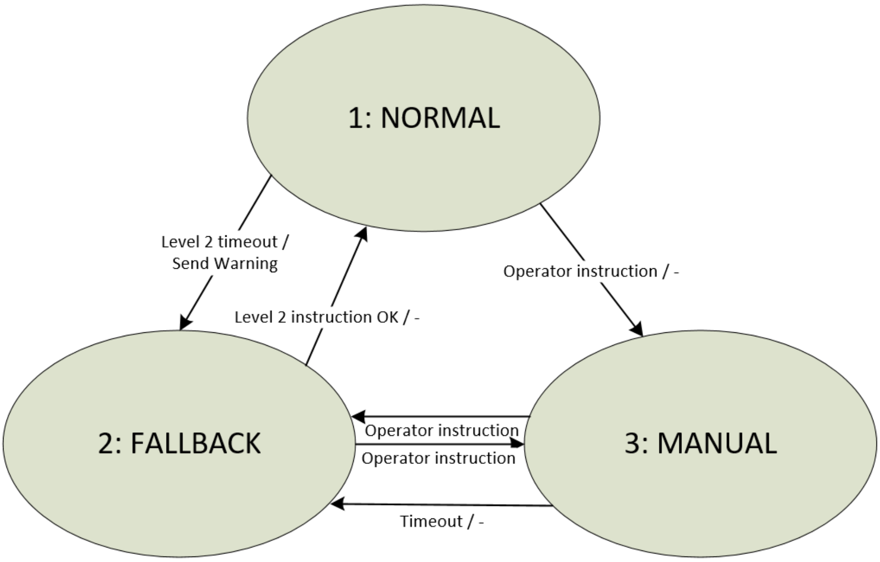


Figure 6.6: Level 1 State Diagram

If the USV is being towed or maneuvered manually, it is advised to select fallback mode 0 so that the USV will continue in a straight line if fallback mode is entered. If the USV is operating in shallow waters, fallback mode 2 is recommended. In fallback mode 2, the USV will start turning in tight circles. Remember that the USV cannot stop if there are waves propelling it. If the USV is operating in the open sea or along the coast, fallback mode 2 with autopilot should be used. A safe direction should be chosen for course angle. In the event of system failure, the USV will then navigate in the safe direction. Fallback mode 2 can also be used to get the USV to the desired position if level 2 becomes permanently lost. The remote operator will have to alter the desired course angle manually. Waypoint navigation will not be implemented in the fallback system as part of the master project.

6.5 Interface

The level 1 system will interface level 2 and the remote operator. For transparency and ease of debugging, it was decided that the interfaces should be human readable. The communication links follow a protocol based on the NMEA0183 standard which is commonly used for marine applications. Several characters are reserved, including the following:

ASCII	Hex	Dec	Use
$	0x21	33	Start delimiter
,	0x2C	44	Field delimiter
*	0x2A	42	Checksum delimiter
<LF>	0x0A	10	End of message

Message format:

$	*MessageID*	,	$Data_1$	,	$Data_2$	,	(. . .) $Data_N$	*	*Checksum*	<LF>

The message field contains five characters for identifying the message. The data fields can contain any character string excluding reserved characters. The checksum is defined as the XOR of all the bytes between the dollar sign and the asterisk. The number is represented with two characters as a hexadecimal value.

Output from Level 1:

Field	Use	Description
$MessageID$	-	"CR601"
$Data_1$	Leak Status	1 = True, 0 = False
$Data_2$	Power Settings	6 character string. Example: "110101"
$Data_{2-1}$	Level 2	1 = Disabled, 0 = Enabled
$Data_{2-2}$	Level 3	1 = Disabled, 0 = Enabled
$Data_{2-3}$	GPS 1	1 = Disabled, 0 = Enabled
$Data_{2-4}$	Iridium 1	1 = Disabled, 0 = Enabled
$Data_{2-5}$	Radar & AIS	1 = Disabled, 0 = Enabled
$Data_{2-6}$	Pumps	1 = Disabled, 0 = Enabled
$Data_3$	Load Power	Power consumed by load [W]
$Data_4$	Panel Power	Power yield from panel 1 & 2 [W]
$Data_5$	Battery	Battery Voltage [V]
$Data_6$	Level 1 state	1 = NORMAL, 2 = FALLBACK, 3 = MANUAL
$Data_7$	Fallback Mode	Selected fallback mode (0, 1 or 2)

Message from Level 2 to Level 1:

Field	Use	Description
$MessageID$	-	"BBB01"
$Data_1$	Rudder Angle	Angle[Deg]$*$10. Min: -450, max: 450.
$Data_2$	Thruster	Min: -100, max: 100. Forward: Positive value.
$Data_3*$	Power Settings	6 character string. Example: "110101"
$Data_{3-1}$	Level 2	1 = Disabled, 0 = Enabled
$Data_{3-2}$	Level 3	1 = Disabled, 0 = Enabled
$Data_{3-3}$	GPS 1	1 = Disabled, 0 = Enabled
$Data_{3-4}$	Iridium 1	1 = Disabled, 0 = Enabled
$Data_{3-5}$	Radar & AIS	1 = Disabled, 0 = Enabled
$Data_{3-6}$	Pumps	1 = Disabled, 0 = Enabled
$Data_4*$	Fallback Mode	Selected fallback mode (0, 1 or 2)

Message from remote operator to Level 1:

Field	Use	Description
$MessageID$	-	"RCC01"
$Data_1$	Rudder Angle	Angle[Deg]∗10. Min: -450, max: 450.
$Data_2$	Thruster	Min: -100, max: 100. Forward: Positive value.
$Data_3*$	Power Settings	6 character string. Example: "110101"
$Data_{3-1}$	Level 2	1 = Disabled, 0 = Enabled
$Data_{3-2}$	Level 3	1 = Disabled, 0 = Enabled
$Data_{3-3}$	GPS 1	1 = Disabled, 0 = Enabled
$Data_{3-4}$	Iridium 1	1 = Disabled, 0 = Enabled
$Data_{3-5}$	Radar & AIS	1 = Disabled, 0 = Enabled
$Data_{3-6}$	Pumps	1 = Disabled, 0 = Enabled
$Data_4*$	Fallback Mode	Selected fallback mode (0, 1 or 2)
$Data_5$	Manual Control	Obtain Manual Control = 1. Else 0.

6.6 Physical Layout

Since components had to be selected prior to the arrival of the USV in order for the project to stay on schedule, the physical system layout in the hull of the vehicle was decided with the aid of FreeCAD software and a model of the USV. Figure 6.7 shows the placement of the cases, the batteries, ADCP and AutoNaut provided motor controller in the hull. The batteries are the only units with significant impact on the USVs balance. Several cases were evaluated for level 1, 2 and 3. The selected where suggested by the workshop at NTNU. The dimensions are 30 x 30 x 17 cm, they are IP67 rated and have transparent lids. The physical layout and mounting of components and devices inside the cases were performed by Peter Knutsen.

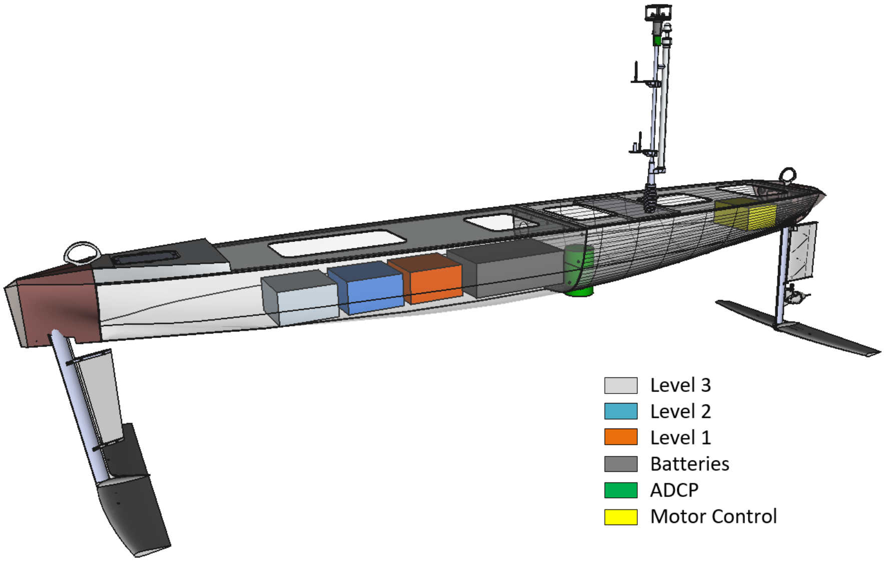


Figure 6.7: Placement of units in the hull.

6.7 Fallback Autopilot

The fallback autopilot will be used in situations where the main navigation system for some reason cannot control the USVs rudder. That could be due to the need for conserving power or due to faults in the level 2 subsystem. Since the autopilot is only intended for fallback purposes, too much time could not be used for its design. However, it is very important that it works.

The only available inputs for the level 1 autopilot are speed over ground (SOG), course over ground (COG) and USV position. The information comes from the level 1 GPS which transmits data at 1 Hz. Since yaw rate is unavailable and derivation of signals can lead to problems, Proportionate Integral (PI) control was chosen. Integral action is needed due to forces from wind and waves.

Since sea trials are resource demanding, a model was made for the initial tuning.

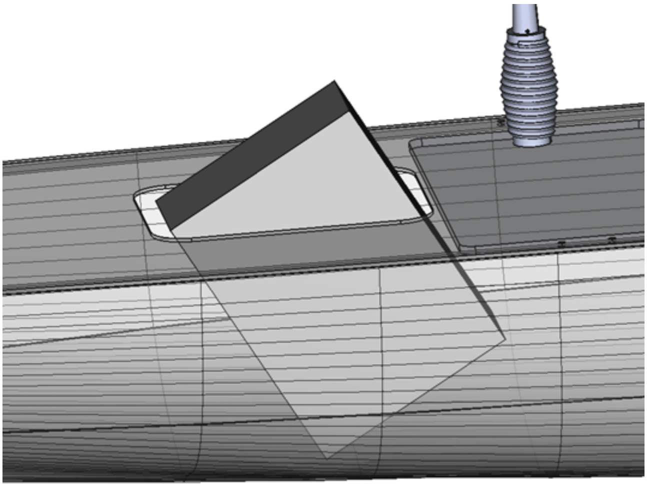


Figure 6.8: Testing if 40 x 30 x 17 cm case would fit using CAD.

Since there was no time for data gathering and maneuvering tests during the delivery of the USV in March, assumptions had to be made when making the model. According to the manufacturer of the USV, it is capable of staying within a 30-meter radius from a point (3). The following assumption was made: At a speed of 1.5 knots ($0.77 m/s$) and a rudder angle of 20^o, the USV will have a turning circle with a radius of 30 meters. In the assumed case, the total distance of a 360^o turn is $2\pi * 30[m] = 60\pi[m]$. The total time of the full turn is given by $distance/speed \implies 60\pi/0.77 = 245[s]$. That yields a yaw rate of $360/245 = 1.47^o/s$

Because the available data were insufficient for tuning an advanced ship model anyway, the first-order Nomoto model was chosen (equation 6.1). The first-order Nomoto model together with an additional integrator, for going from rudder angle to heading angle, is used in most commercial autopilot systems (21). δ is the rudder angle, r is the yaw rate, ψ is the yaw angle and K and T are constants.

$$\frac{r}{\delta}(s) = \frac{K}{(1 + Ts)} \tag{6.1}$$

$$\frac{\psi}{\delta}(s) = \frac{K}{s(1 + Ts)} \tag{6.2}$$

Equation 6.1 is the transfer function from rudder input to yaw rate. The model can be tuned to the assumed system dynamics by setting the K and T. K can be found by using the final value theorem which states:

$$\lim_{t \to \infty} f(t) = \lim_{s \to 0} s * F(s) \tag{6.3}$$

It has been assumed that the stationary yaw rate, at a rudder angle $\delta = 20^o$, is $r = 1.47\ ^o/s$. By using the final value theorem (6.3), an expression for the stationary value of yaw rate can be derived:

$$\lim_{t \to \infty} r(t) = \lim_{s \to 0} s * \frac{K}{1 + Ts} * \delta(s) = \lim_{s \to 0} s * \frac{K}{1 + Ts} * \delta(s) \tag{6.4}$$

$$\delta(s) = \delta * \frac{1}{s} \implies \lim_{t \to \infty} r(t) = K * \delta \tag{6.5}$$

$$\implies K = \frac{1.47}{20} = 0.0736 \tag{6.6}$$

The time constant T is the time for when the transfer function (6.1) has obtained 63 percent of its stationary value. Based on the observance of the USV during delivery, T was set equal to 1. This guess had to be made since there was no available data. The resulting first-order model for yaw rate:

$$\frac{r}{\delta}(s) = \frac{0.0736}{(s + 1)} \tag{6.7}$$

The transfer function was used in a USV model implemented in Simulink. Figure 6.9 shows the Simulink implementation of equation 6.7. Note that a saturation block is placed on the rudder input. The block simulates the physical constraints of the USV by limiting the signal to $\pm 45^o$.

As stated, a PI controller was chosen for the course over ground control. In order to avoid the detrimental effects of integrator wind-up, an anti-windup mechanism was

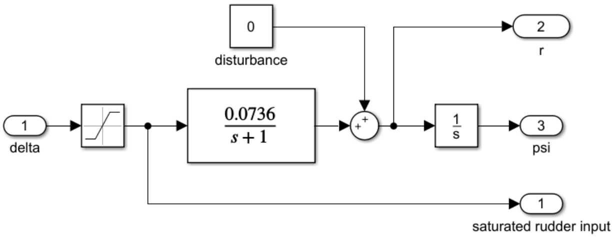


Figure 6.9: Simulink heading model for USV

added. In Simulink, anti-windup can be added to a controller by simply checking "limit output" in the PID advanced tab in the settings for the PID-block. "Back-calculation" was selected as method in the Simulink block as it was the method chosen for implementation of the anti-windup in the real system.

For increased realism, a zero-order hold block and a transport delay block was added between the heading output δ in the heading model and the input for the PI controller. The zero-order hold samples the input signal at a 1 Hz frequency and outputs a zero-order function (constant value) that represents the input signal. The transport delay block delays the signal from input to output. A two-second delay was selected. This represents the real system because the GPS used outputs COG with a frequency of 1 Hz.

Spending a lot of time tuning the PI controller in Simulink was not prioritized because the assumptions for tuning the USV model were relatively questionable. It was found that $K_p = 1$ and $K_i = 0.1$ yielded satisfactory results. Therefore it was decided to use these values as gain parameters in the initial sea trials. When testing the gain parameters, a constant disturbance was added to yaw rate to ensure that the integral action in the PI controller was sufficient to tackle wind and waves.

Figure 6.11 shows the result of the simulation. Note that a step input is inputted to the controller. The initial heading of the USV is 0^o, and the desired heading is 90^o. The rudder angle is saturated at 20^o until the USV reaches the desired heading angle. As soon as the desired heading angle has been obtained (after roughly 45s), the rudder

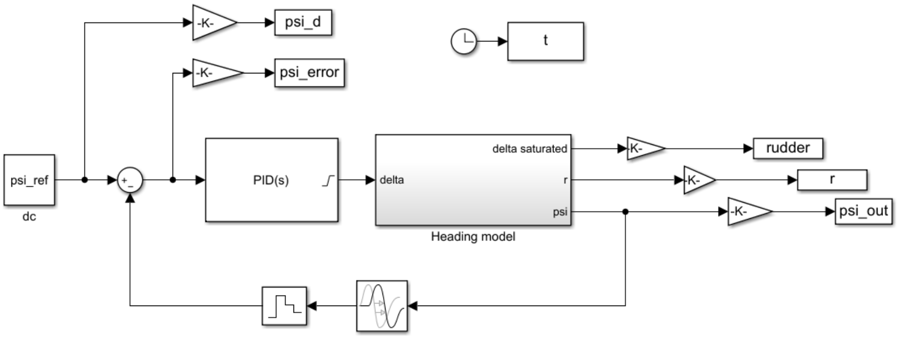


Figure 6.10: Simulink model for USV with PI controller

angle starts decreasing, that shows that integrator wind-up has not occurred. Note that the rudder angle after 200 seconds is -8^o, while heading angle is constant. This is due to the added disturbance on yaw rate and shows the integral action in the controller.

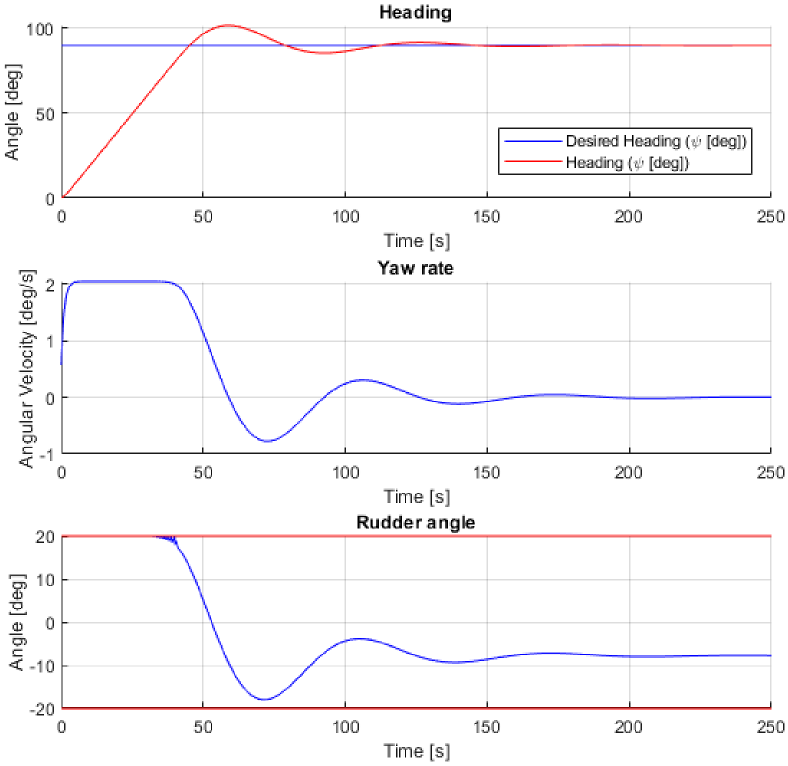


Figure 6.11: Simulation heading control with disturbance on yaw rate.

Chapter 7

Implementation

This chapter covers the implementation of the desired system behavior for level 1. It starts by introducing the different devices and related considerations, then moves to actual implementation. The notation used in the CR Basic user manual was adhered to during code implementation.

7.1 Campbell Scientific C6 Computer

The Campbell Scientific Cr6 computer was selected for level 1 in semester one due to its proven reliability. Campbell Scientific delivers robust computers for data sampling and control. Their computers have been successfully used for projects in cold environments such as long-term monitoring in Greenland (22), maritime environments with movement and vibrations such as a buoy system in the Caribbean Sea (23) and in systems with strict reliability requirements such a weather stations at military airports (24). The Campbell Scientific CR6 uses a Renesas RX63N processor with a clock rate of 100 MHz and has 16 general I/O pins with dedicated hardware for support of numerous communication protocols (25). The CR6 was chosen over other similar products from Campbell Scientific due to the need of I/O ports. The CPU in the CR6 is also among the fastest in their product range.

Implementation of software on the CR6 is done with the use of PC400 Datalogger Support Software and CR Basic Editor. The PC400 Datalogger support software serves several functions. First of all, it enables the user to send program code and retrieve data from the unit via USB interface with a Windows computer. It also lets the user set the internal clock of the CR6 easily. With the Windows program, publicly declared variables can be read in real time from the device via the USB cable to ease the process of debugging. Unfortunately, the software does not support debugging with the use of common tools such as breakpoints and memory access. The lack of such tools does increase development time significantly and should be taken into account when planning development of future system functions.

The CR Basic editor lets the user write programs for the CR6 device. The editor has a built-in compile function that will check for faults such as syntax errors and incorrect memory allocation, but the program is recompiled on the CR6 device after it has been transferred to the device. The file extension of the programs that are sent to the CR6 is ".CR6".

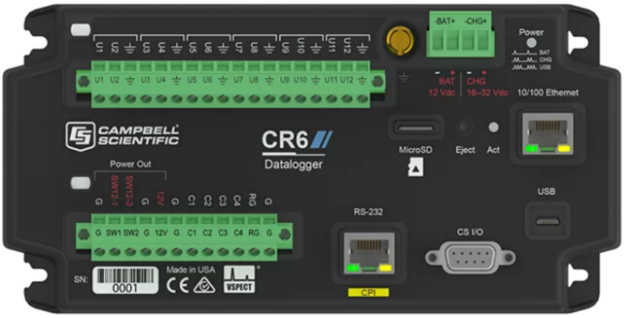


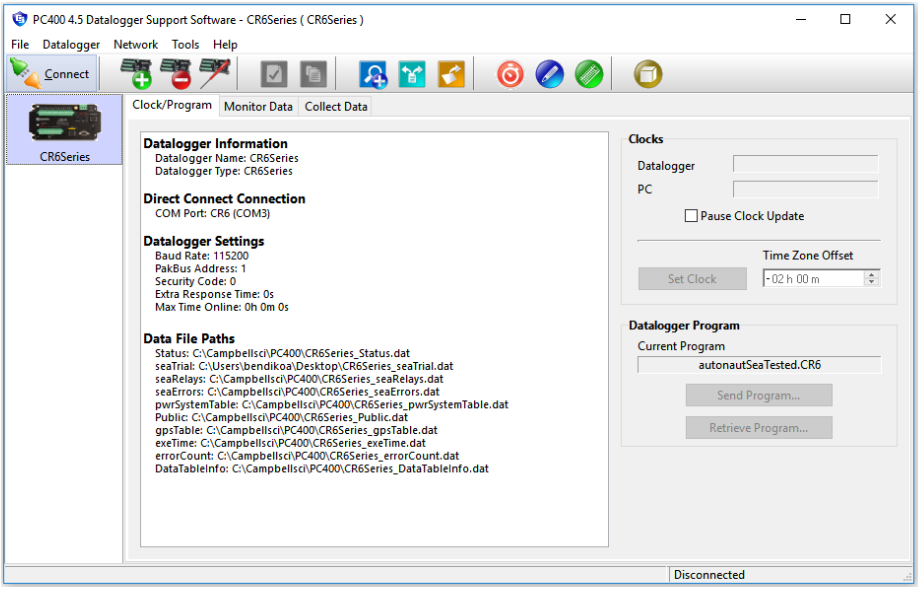
Figure 7.1: Campbell Scientific CR6

Device	Campbell Scientific CR6
Unit Cost	18 500 NOK excl. SW
Operating System	CR6 OS
Processor	Renesas RX63N 100 MHz
Input Voltage	10 - 18 V DC
Temperate Rating	-40 ^{o} to +70 ^{o}

The PC400 Datalogger Support Software can be seen in figure 7.2. Note that the window has three tabs. "Clock/Program" provides information about the connected computer and communication port settings. The "Monitor Data" tab enables real-time view of the publicly declared program variables, which is useful for debugging. The "Collect Data" tab is used for retrieving data tables that are stored on the device during operation. The blue icon on the menu is a shortcut to the CR6 editor. In order to send a program to the CR6, the device must first be connected by selecting "Connect," then the "Send Program..." button must be selected. A file selector window will then be opened and a .CR6 file can be selected. The CR6 will automatically attempt to compile the program and run it when transfer is complete. Note that all stored data tables on the CR6 are deleted when a new program is compiled. Therefore one must remember to retrieve any files prior to sending a new program.

7.2 CR Basic

There are two methods for implementing the desired behaviour on the CR6. One method is to use a program called *Short Cut*, the other option is to use the *CR Basic Editor*. *Short Cut* can be used for simple tasks such as sampling a sensor at a set interval and storing the data, *CR Basic Editor* is the only suitable alternative for the USV system. The CR Basic programming language is based on BASIC, which is a programming language that was released in 1964 (26). Although CR Basic has functionality that was seen in the original BASIC language, it is a lot more similar to newer procedure-oriented dialects of BASIC. The operating system (OS) on the CR6 automatically runs



```
Datalogger Information
   Datalogger Name: CR6Series
   Datalogger Type: CR6Series

Direct Connect Connection
   COM Port: CR6 (COM3)

Datalogger Settings
   Baud Rate: 115200
   PakBus Address: 1
   Security Code: 0
   Extra Response Time: 0s
   Max Time Online: 0h 0m 0s

Data File Paths
   Status: C:\Campbellsci\PC400\CR6Series_Status.dat
   seaTrial: C:\Users\bendikoa\Desktop\CR6Series_seaTrial.dat
   seaRelays: C:\Campbellsci\PC400\CR6Series_seaRelays.dat
   seaErrors: C:\Campbellsci\PC400\CR6Series_seaErrors.dat
   pwrSystemTable: C:\Campbellsci\PC400\CR6Series_pwrSystemTable.dat
   Public: C:\Campbellsci\PC400\CR6Series_Public.dat
   gpsTable: C:\Campbellsci\PC400\CR6Series_gpsTable.dat
   exeTime: C:\Campbellsci\PC400\CR6Series_exeTime.dat
   errorCount: C:\Campbellsci\PC400\CR6Series_errorCount.dat
   DataTableInfo: C:\Campbellsci\PC400\CR6Series_DataTableInfo.dat
```

Time Zone Offset: - 02 h 00 m

Current Program: autonautSeaTested.CR6

Figure 7.2: Campbell Scientific CR6

the program when the device is turned on. Only one program can run, and a program can only have one thread. Hence, the OS does not have the characteristics associated with a real-time operating system (RTOS). That does restrict the number of options for implementation of the system functions, but it could be positive for future developers who are not accustomed to the particularities of RTOS.

A CR Basic program consists of one file only and has the following form:

1. Definition of constants
2. Declaration of global variables
3. Definition of data tables for logging of data
4. Definition of user functions
5. Program code that runs at power on

6. Program code that runs sequentially at fixed frequency

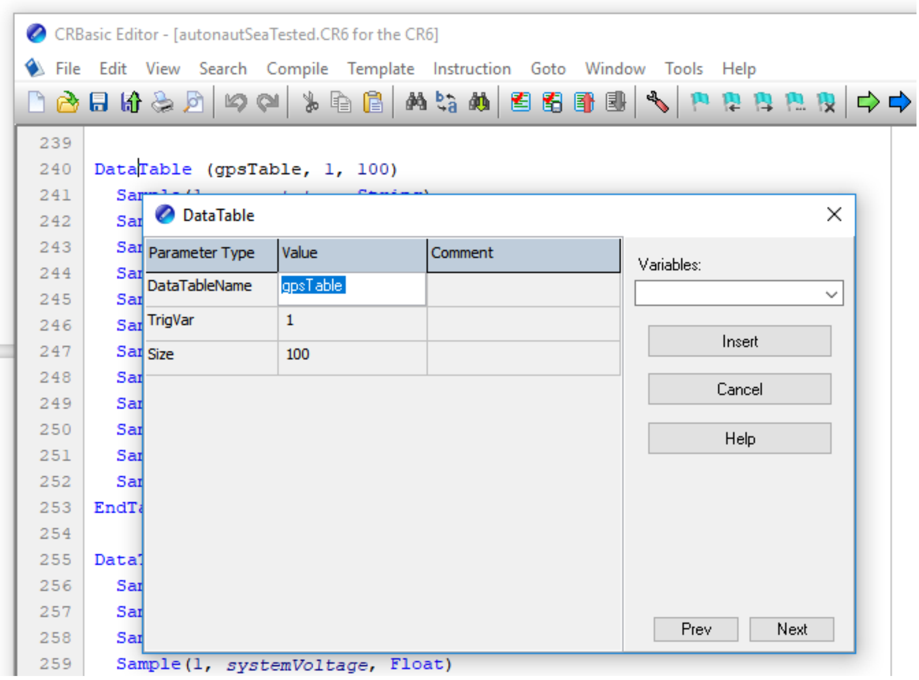
CR Basic supports programming functionality such as Loops, conditional statements, and functions. The language does not support object-oriented functionality and does not let the user define classes or even structures as found in c++ and c. User-defined functions can only return one variable unless arrays are used as return type and can only be nested to a maximum of two levels deep. Since deeper nesting is unsupported, small sub-functions that are useful to increase the readability of the code cannot be used in all cases, resulting in less reuse of code. On the positive side, the CR Basic programming language has several built-in functions that are useful for system implementation such as functions for parsing data or reading serial buffers.

It is important to note that there are two different compile modes to choose from: Sequential mode and pipeline mode. In sequential mode, the program runs one thread sequentially. The thread is restarted at a specified time interval. The restart occurs even if all the program code did not complete. If the computational requirements to the program sequence for some reason increases during runtime, the last computations in the program will not execute because the thread restarts.

The alternative to sequential mode in CR Basic is pipeline mode. In pipeline mode, the program will sample incoming data at a given interval, as it does in sequential mode. However, if the processing of data cannot be completed within the time frame, it will be stored in a buffer, and an attempt will be made to process after the next data sampling instance.

Pipeline mode is well suited for simple applications where data only has to be sampled, process and stored. For example, a system that samples analog sensors and calculate scientific values from the inputs. For a more complicated system that includes conditional expressions and several system states, the tradeoff by using pipeline mode is that the system becomes prone to race conditions. This can occur even if there is only one processor because the operating system will perform task switching within the program. Due to the potential dangers and unpredictability that become present in the event of race conditions, pipeline compilation mode should not be used for the USV system. For simple data logging purposes, however, pipeline mode is ideal.

Figure 7.3 shows an extract from the CR Basic Editor software. The functionality



```

DataTable (gpsTable, 1, 100)
  Sa
  Sa
  Sa
  Sa
  Sa
  Sa
  Sa
  Sa
  Sa
  Sa
  Sa
  Sa
EndT

Data
  Sa
  Sa
  Sa
  Sample(1, systemVoltage, Float)
```

Parameter Type	Value	Comment
DataTableName	gpsTable	
TrigVar	1	
Size	100	

Figure 7.3: CRBasic Editor

is quite limited, but a some points should be noted. The blue icon with a red check-symbol on the menu line is the shortcut for compile. The program will be recompiled on the CR6 after it is transferred, but by checking for syntax errors prior to transfer time can be saved. The flags are for setting bookmarks in the code. If a word market in blue is right-clicked in the code, a window will pop up and show the expected input arguments as shown in figure 7.3.

7.3 Garmin GPS16x HVS

The Garmin GPS16x HVS (27) was selected as it is IPX7 rated and has a built-in antenna. Few commercial of the shelf devices meet those criteria. An alternative would be to choose a GPS to be placed inside the IP67 rated case in the hull, but water resistant antenna connectors were not found. The device can be replaced at a later stage if GLONASS or Galileo is required. The device is interfaced with messages that follows the NMEA0183 protocol. Baud rate is not defined in the NMEA standard. The baud rate can be set to 4800, 9600, 19200 or 38400. Lower baud rate increases immunity against radio frequency interference (RFI) but requires more time for transfer. 38400 was selected and tested. The device follows RS-232 defined voltage levels but also supports TTL voltage levels. The device does not support polling but outputs the selected messages at a frequency of 1 Hz. The user can enable or disable the different available messages. The available NMEA-approved output sentences are: GPALM, GPGGA, GPGLL, GPGSA, GPGSV, GPRMC, GPVTG. In addition, the device can provide the following Garmin proprietary sentences: PGRMB, PGRME, PGRMF, PGRMM, PGRMT, and PGRMV. In order to be able to meet system requirement *A.REQ.2.06.4*, GPRMC was enabled. GPRMC provides SOG, COG, and position. The GPGGA message enabled in addition because it provides the following useful information: Signal quality, number of satellites with fix and horizontal accuracy of position. That information is necessary to meet requirement *B.REQ.2.08.1* - Device Error Monitoring. Another useful feature in the device is its ability to enter sleep mode. In standby mode, the current consumption is extremely low ($< 10\mu A$). In normal mode of operation, the current consumption is around 65 mA at 12 V. If many messages are enabled, the current consumption increases due to increased duration of voltage excitation on the serial line. The data acquisition time is stated to be approximately 45 seconds from cold start (27).

Device	Garmin GPS16x HVS
Unit Cost	1100 NOK
Interface	NMEA0183
Input Voltage	8 - 40 V DC
Temperate Rating	-30 ^{o} to +80 ^{o}
Protection	IPX7

Figure 7.4: Garmin GPS16x HVS

7.4 RockBLOCK+ Iridium

The RockBLOCK device is IP68 rated and hosts an Iridium 9602 unit, an antenna and a voltage regulator. The 9602 unit is interfaced with RS-232 using AT commands. Depending on the conditions, the unit can send or receive a message approximately every 40 seconds. The cost is 0.1 to 0.05 GBP per 50-byte message, depending on the bundle that is bought. In addition, there is a 10 GBP line rental per month. The system has a limited bandwidth and is therefore only suitable for simple control monitoring or tracking applications. The maximum package sizes are 340 bytes for sending and 270 bytes for receiving. Although the latency is typically a few seconds, it may be as much as a minute or more.

The RockBlock device has a built-in voltage regulator and can be powered from a source delivering from 9 V to 30 V. In the vehicle the device will be powered with 12 V. The idle current at 12 V when the device is awake, is 16 mA, but in order to transmit, the device uses a built-in energy storage that requires charging. It takes up to 20 seconds to reach full charge, and the consumption is 225 mA at 12V during this period. The device features a sleep mode at which the consumption is $20 \mu A$ at 12 V. It is beneficial to keep the device in this mode instead of turning off the power as the

Figure 7.5: RockBLOCK+

internal charge is then maintained. Since the internal charge is maintained, the device current draw will not fluctuate as much, and it will be able to transmit immediately when instructed to. If the power to the device is to be activated and deactivated with a relay, it is important to remember that the device might require 20 seconds to charge and 1 minute to transmit, a total of 80 seconds.

Device	RockBLOCK+
Unit Cost	2300 NOK
Interface	AT commands
Input Voltage	9 - 30 V DC
Temperate Rating	-40 ^{o} to +85 ^{o}
Protection	IP68

The communication interface to the device consists of serial communication at RS-232 voltage levels. The baud rate is configurable. The device uses 8 data bits and 1 stop bit. In addition to the serial interface, the device has a ring alert signal wire, a network availability signal line and a sleep mode line. Only the sleep mode line will be used in the system.

When the Campbell Scientific C6 computer sends a message to the RockBlock device, the message is loaded into the so-called mobile originated message buffer in the device. A message transfer session between the RockBlock and the gateway SBD subsystem is then initiated. The RockBlock will notify the CR6 on the success of the message. If the RockBlock receives a message, it will be stored in the mobile terminated buffer in the device and can be transferred to the CR6 via the serial interface.

The communication between the Iridium unit in the rock block and the CR6 level 1 computer follows an Iridium proprietary protocol that is described in a document called ISU AT Command Reference. A total of 180 commands are described, in which roughly 50 are supported by the Iridium 9602 transceiver. Out of those commands, 20 concern the handling of short burst data service. After assessing the different functions and possibilities, the following instructions were selected for implementation:

- SBDI (Short Burst Data Initiate)
- SBDRT (Short Burst Data Read Text)
- SBDRB (Short Burst Data Read Binary)
- SBDS (Status)
- SBDTC (Transfer MO to MT)
- SBDWB (Write binary)
- SBDWT (Write text)

By combining the commands listed, the *A.REQ*.2.06 requirements could be met.

7.5 433 MHz Radio Transceiver

The 433 MHz Radio Transceiver requires a 12 V input and is interfaced with RS-232 at TTL voltage levels. Device settings can be changed with a set of AT commands, but in operation, the device functions as a transparent serial link.

7.6 Victron BlueSolar MPPT Controller

The Victron Blue Solar MPPT Controller is an maximum power point tracking (MPPT) controller that optimizes the impedance seen from the PV panel for maximum yield. The device has a battery input, a PV panel input, and a load output. The battery input and load output are connected inside the device, but current is measured separately so that consumption can be estimated. Additionally, the device has a port for communication interface. The communication link is called *VE.direct*. The device is not water resistant, but as seen in figure 6.4, it is mounted inside an IP67 rated case. As explained in Chapter 6.2, two BlueSolar devices are used.

Figure 7.6: Victron Blue Solar MPPT 75|10

Device	Victron BlueSolar MPPT 75|15
Unit Cost	1000 NOK
Interface	VE.Direct (RS-232 voltage levels)
Temperate Rating	-30 ^{o} to +60 ^{o}
Protection	IP22

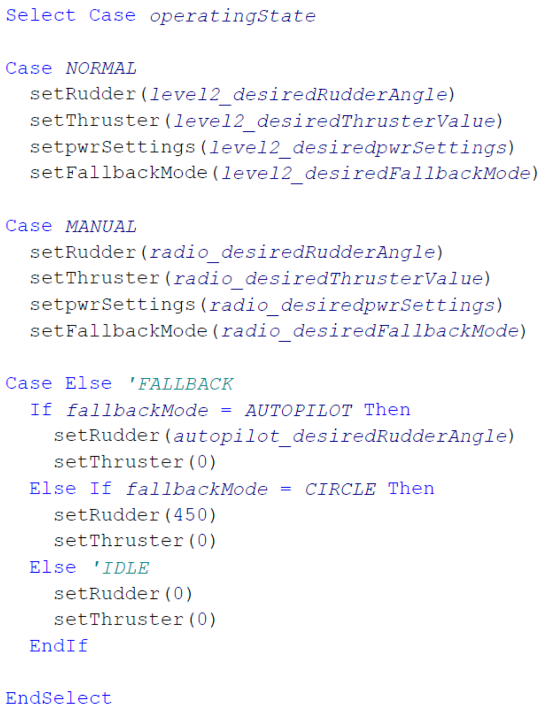
Victron has developed a communication protocol that is intended for connecting their devices. However, they share the protocol so that their devices can be interfaced with other systems too. There are two VE.Direct protocols. One protocol is in human readable text, while the other protocol is in HEX format. The HEX protocol was chosen for two reasons. Firstly it has more functionality, secondly, it requires less data for representing the same information and therefore requires less use of the serial line which in turn reduces power consumption slightly. Data communication with Victron products is described in a whitepaper by Victron (28). The VE.Direct protocol has eight basic commands. The most important are the "Get" and "Set" commands. The "Get" command lets the user retrieve data from the registers on the controller, whilst the "Set" command is used to set device parameters. There are over 100 registers that can be set or acquired based on desired functionality. After a thorough assessment of the possibilities, the following commands were picked for implementation in the system:

1. Get device state
2. Get load current
3. Get panel power
4. Get charger error code
5. Set relay control register

Command 1 and 4 are used for device error monitoring (*B.REQ.2.08.1*), command 2 and 3 are used to achieve load and PV panel power monitoring (*B.REQ.2.11.2* and *B.REQ.2.11.3*), and command 5 was used to configure the built-in relay. The relay was set to always be on.

7.7 Program Polling Loop and State Machine

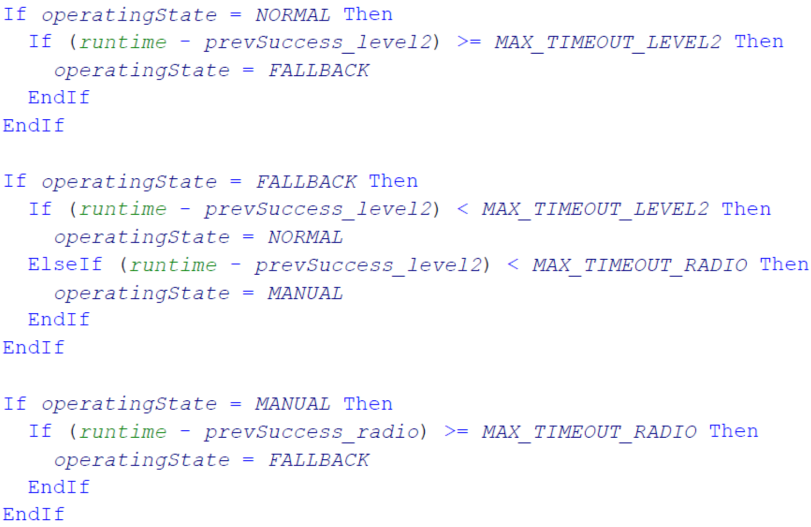
The program was made using PC400 software as described in Chapter . The core part of the program is a polling loop that will execute indefinitely at a one-second interval as long as the CR6 is powered. The repeatedly executed code sequence contains a state machine in accordance to the desired behaviour shown in figure 6.6. The state machine is implemented using the switch case implementation model, which is one of the most common implementation approaches (29). The executing sequence consists of three main stages in the. Stage one is to obtain system information, stage two is to set the system state and control actuators based on the acquired information, and stage three is to output data. An extract from the program code is seen in figure 7.7.



```
Select Case operatingState

Case NORMAL
  setRudder(level2_desiredRudderAngle)
  setThruster(level2_desiredThrusterValue)
  setpwrSettings(level2_desiredpwrSettings)
  setFallbackMode(level2_desiredFallbackMode)

Case MANUAL
  setRudder(radio_desiredRudderAngle)
  setThruster(radio_desiredThrusterValue)
  setpwrSettings(radio_desiredpwrSettings)
  setFallbackMode(radio_desiredFallbackMode)

Case Else 'FALLBACK
  If fallbackMode = AUTOPILOT Then
    setRudder(autopilot_desiredRudderAngle)
    setThruster(0)
  Else If fallbackMode = CIRCLE Then
    setRudder(450)
    setThruster(0)
  Else 'IDLE
    setRudder(0)
    setThruster(0)
  EndIf

EndSelect
```

Figure 7.7: Behaviour implementation using switch case

The system monitoring is time driven. The program will check system devices at selected time intervals. This is because the CR6 does not support interrupt driven actions and it cannot constantly poll the connected devices using multi-threading. Activating the other level 1 devices increases the system power consumption. Active serial communication increases the power consumption with approximately 20mA and having a device constantly turned on is costly. In the implemented program, the schedule for reading and writing to external devices is kept by using a system counter that keeps track of seconds since the system was turned on. An example of use of the counter is seen in the code section that alters system state in the event of timeouts. Figure 7.8 shows an extract from the code implementation that is in accordance with the state transitions as defined in figure 6.6. If the system is in *NORMAL* operating mode, meaning that the desired rudder and thruster values from the level 2 navigation system are used, it will enter *FALLBACK* mode if the time since last successfully received message from level 2 has exceeded the max timeout threshold (*MAX_TIMEOUT_LEVEL2*). If the *operatingState* is *FALLBACK* and the time since last instruction from level 2 was received under less than the maximum timeout, the system will enter *NORMAL* and so on.



```
If operatingState = NORMAL Then
  If (runtime - prevSuccess_level2) >= MAX_TIMEOUT_LEVEL2 Then
    operatingState = FALLBACK
  EndIf
EndIf

If operatingState = FALLBACK Then
  If (runtime - prevSuccess_level2) < MAX_TIMEOUT_LEVEL2 Then
    operatingState = NORMAL
  ElseIf (runtime - prevSuccess_level2) < MAX_TIMEOUT_RADIO Then
    operatingState = MANUAL
  EndIf
EndIf

If operatingState = MANUAL Then
  If (runtime - prevSuccess_radio) >= MAX_TIMEOUT_RADIO Then
    operatingState = FALLBACK
  EndIf
EndIf
```

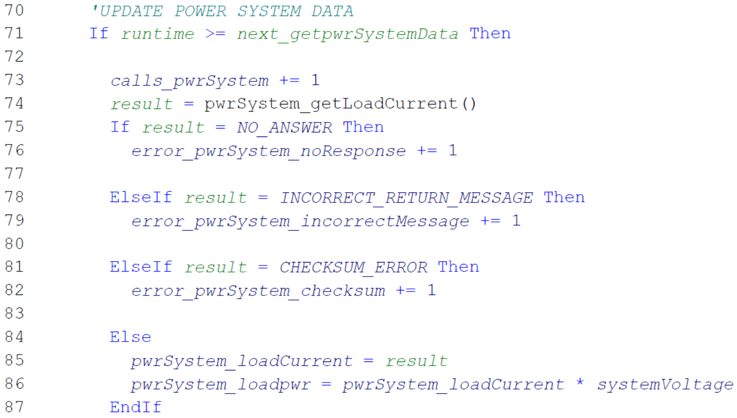
Figure 7.8: Conditional statements for state transitions.

Requirement *B.REQ.2.08.1* concerning error monitoring was implemented so that the information could be used for two purposes. Firstly the implemented error monitoring allows for the system to make autonomous decisions based on the errors occurring in the system and secondly to provide the remote operator with accurate information of the internal state of the system. Autonomous decisions based on internal errors were not implemented because it was seen as unwise to increase the level of complexity prior to sea testing. The only automatic changes of behaviour in level 1 are the ones described in figure 6.6 in Chapter 6. Examples of functionality that would be very easy to implement in the provided framework are: Disabling level 2 and 3 if the excepted energy yield is insufficient for prolonged operation or rebooting of level 2 in the event of error detection. Providing information to the operator, on the other hand, is also seen as very important. According to an influential book on the topic on system safety, which describes the challenges related to the role of humans in automated systems (15), it is vital to provide the operator with information sufficient for creating a clear picture of the system state. Therefore the *B.REQ.2.08.1* was prioritized during implementation.

7.8 Error Monitoring

For every device call that is made in the system, the result of the call is logged. If an error occurred, the implemented function will return error and in most cases also the reason for the error. An error table is stored and can be sent to the operator at user-defined time intervals. With the current implementation, it is also possible to add functionality for sending warnings in the event of errors or to send the error table when a certain number of errors is reached. It is also easy to add functionality for automatic action in the event of errors.

Figure 7.9 shows an example of how the return value of the implemented functions are used for logging the internal state of the level 1 system. The code segment is part of a sequence that uses functions implemented for acquiring data from the Vectron BlueSolar MPPT controller. Line 71 is a conditional statement that checks if it is time to obtain new data from the device. Line 73 increments the variable holding track of the number of calls made to the power system device. Line 74 calls the function *powerSystem_getLoadCurrent()* and saves the returned value in the variable *result.* The return type for the function is an integer. Line 75, 78 and 81 checks if the implemented



```
70      'UPDATE POWER SYSTEM DATA
71      If runtime >= next_getpwrSystemData Then
72
73        calls_pwrSystem += 1
74        result = pwrSystem_getLoadCurrent()
75        If result = NO_ANSWER Then
76          error_pwrSystem_noResponse += 1
77
78        ElseIf result = INCORRECT_RETURN_MESSAGE Then
79          error_pwrSystem_incorrectMessage += 1
80
81        ElseIf result = CHECKSUM_ERROR Then
82          error_pwrSystem_checksum += 1
83
84        Else
85          pwrSystem_loadCurrent = result
86          pwrSystem_loadpwr = pwrSystem_loadCurrent * systemVoltage
87        EndIf
```

Figure 7.9: Use of error returns from implemented function.

function returns an error. The *powerSystem_getLoadCurrent()* function can return the following errors: *"no answer, "incorrect return message" and "checksum error"*. The errors were defined during implementation of the *powerSystem_getLoadCurrent()* function. If one of the errors occurred, the variables associated with the returned error will be incremented (line 76, 79 and 82). Other measures could be taken by adding code on lines 77, 80 or 83. If none of the error values were returned, success is assumed, and the return value is used for its purpose. The errors that are returned are defined as constants and do not hold integer values that could match an actual return value.

The error table that contains the count for all the internal system errors can be extended in the future if more functions are added to the program. Currently, the table logs the following parameters:

Variable name	**Description**
errors_resetTime	Last time the error register was reset
calls_level2	Communication attempts with level 2 system
calls_garmin	CR6 input buffer for GPS device was read
calls_iridium	Communication attempts with RockBlock+ Iridium
calls_pwrSystem	Communication attempts with Victron BlueSolar
error_level2_noResponse	Level 2 did not acknowledge
error_level2_checksum	Checksum in level 2 message was incorrect
error_level2_incorrectReturnMessage	Level 2 responds with wrong message
error_garmin_noResponse	No data in CR6 input buffer for GPS
error_garmin_checksum	NMEA0183 checksum in incoming message wrong
error_garmin_device	Garmin GPS reports internal error (incl. no sat. fix)
error_iridiumWrite_incorrectMessage	Iridium device returns unexpected message
error_iridiumRead_incorrectMessage	Iridium device returns unexpected message
error_iridiumRead_noResponse	Iridium device is unresponsive (no output)
error_pwrSystem_noResponse	BlueSolar device is unresponsive (no output)
error_pwrSystem_checksum	BlueSolar device message fails checksum test
error_pwrSystem_incorrectMessage	BlueSolar device returns unexpected message

7.9 Device Interfacing

The process of implementing the different devices to the CR6 computer consisted of several steps. Initially, the user manuals had to be assessed to understand how the functionality that was promised on the various websites could be used in practice. By reading the manuals, a set of desired functions could be selected for further pursuit. After that, the physical connection between the device at hand and the CR6 computer had to be established. This step was also time-consuming because each manufacturer seem to use different colors for wires, different voltage levels, different logic high/low, different default baud rates, stop bits, etc. for their serial interface protocols. Once a physical connection was established functions were written to allow for sending instructions to the devices and reading the required data from the them. A total of nine functions were implemented for interfacing the devices connected to the CR6 computer:

Function name	**Description**
NMEA0183_ChecksumOK(input) *(Sub function)*	Compares checksum provided at the end of NMEA0183 message with the calculated checksum of message. Return: True / False.
setRudder(angle)	Sets the rudder servo to desired angle. Input range: -450 to 450 (integer value). Returns ERROR if desired angle is outside physical constraints of servo.
setThruster(power)	Sets thruster power. Input range: -100 to 100 (integer value). Negative values for reverse. Returns ERROR if desired value is outside limits.
radio_readWrite()	Transmits system information on serial port connected to 433 MHz radio, parses message on input buffer and updates desired values for use in manual control mode. Returns NO_ANSWER, CHECKSUM_ERROR or SUCCESS.
level2_readWrite()	Transmits system information to level 2, parses returned message and updates desired values for use in "normal mode". Returns NO_ANSWER, CHECKSUM_ERROR, INCORRECT_RETURN_MESSAGE or SUCCESS.
garmin_read()	Reads the two most recent messages on the serial buffer dedicated for the Garmin GPS device. Parses GPRMC and GPGAA message and updates GPS data table. Returns NO_ANSWER, INCORRECT_RETURN_MESSAGE, CHECKSUM_ERROR, DEVICE_ERROR or SUCCESS.

Function name	**Description**
iridium_messageOK(pointerToMessage) (*Sub function*)	Checks message sent from iridium device. Returns TRUE or FALSE.
iridium_writeToOutbox(message)	Transfers message from CR6 to Mobile Originated (MO) buffer in iridium device. Returns ERROR or SUCCESS.
iridium_readFromInbox(pointerToMessage)	Checks status of Mobile Terminated (MT) buffer in iridium device, if there is a message waiting, the message is read and the buffer is cleared. Returns INCORRECT_RETURN_MESSAGE, BUFFER_EMPTY, ERROR_READING_BUFFER (or other) or SUCCESS.
iridium_CopyOutboxToInbox()	Copies MO buffer to MT buffer. For the purpose of testing functionality without using Iridium subscription service. No return.
iridium_clearBuffer(selectedBuffer)	Clears selected iridium buffer. Input: 0, 1 or 2. Outputs errors or SUCCESS.
iridium_send()	Initiates satellite session and transmits content of Mobile Originted buffer. Outputs errors or SUCCESS.
iridium_receive()	Initiates satellite session and downloads messages to Mobile Terminated buffer. Outputs errors or SUCCESS.
iridium_signalQuality()	Returnes satellite signal strength or errors.

Function name	**Description**
pwrSystem_checkSumOK(message) (*Sub function*)	Verifies that the checksum of the messages from the BlueSolar device are correct.
pwrSystem_getLoadCurrent()	Obtains load current from BlueSolar device. Returns error or current in amps.
pwrSystem_getDeviceState()	Obtains the device state of the BlueSolar. Returns device state or error codes: CHECKSUM_ERROR, NO_ANSWER or INCORRECT_RETURN_MESSAGE
pwrSystem_getPanelPwr()	Obtains PV panel yield from BlueSolar device. Returns error or power yield in watts.
pwrSystem_getChargerErrorCode()	Obtains BlueSolar error state.
pwrSystem_loadRelayEnable(enableRelay)	Enables or disables the built in load output relay in the BlueSolar device. Only used for testing. May be useful for implementation of functionality in the future. Returns error codes if unsuccessful.

A total of twenty functions were implemented to interface the devices connected to the CR6 computer that controls level 1. The implementation for the different devices varied due to differences in protocols and functionality. The Garmin GPS device does not support polling of data. The device transmits the desired output messages once every second. Therefore the GPS serial port in the CR6 must be activated a minimum of 1 second prior to intended reading of buffer. In order to stay on schedule, functionality to support activation and deactivation of the serial port was not prioritized in the current implementation. At the current sampling frequency (1 Hz) no power would be saved since the serial port would remain. If the functionality is desired in the future, it can easily be added in the section of the polling loop that controls the sleep pin for the GPS. The garminRead function reads the newest GPRMC and GPGAA messages in the buffer. The device uses a 38400 as baud rate, logic 1 is low, 8 data bits and one stop bit.

Five wires are required: Ground, power, TX, RX, and sleep.

The iridium device uses a baud rate of 19200, logic 1 is low, there are 8 data bits and one stop bit. Five wires are used: Ground, power, TX, RX, and sleep. Unfortunately, the AT command protocol does not support checksum, but the return message contains the transmitted instruction and ends with "OK", so checks are implemented to check those indicators. Message content that is sent on the satellite link can, of course, contain user defined checksum. That has not been implemented, because Iridium perform error checks that are hidden for the user. However, the NMEA0183 checksum function can easily be used for the purpose if desired.

The BlueSolar device uses a baud rate of 19200 and logic 1 is defined as high. The VE.Direct interface cable had custom connectors in both ends, so one connector was removed so that the wires in the cable could be connected directly to the input and outputs of the CR6. The color coding for the wires in the cable is not provided in the manual, but by using the trial and error method, it was found that the black wire is power, the red wire is ground, the white wire is device RX, and green wire is device TX. Note that the black wire (power) is not required for the communication interface when the BlueSolar is connected to batteries.

The motor controller box delivered by AutoNaut Ltd. is controlled with Pulse Width Modulated (PWM) signals and does not provide feedback. The CR6 has built in functions for outputting PWM signals, but functions for converting a desired rudder angle and thruster power value to PWM signals had to be created.

7.10 Leak detection

Requirement *B.REQ.2.08.3*, leak detection, was implemented by using the pwrSystem_getLoadCurrent function. When leak check is initiated, the load current is measured for a set time, and the average load current is calculated. Then the bilge pumps are activated by enabling an output pin that controls the relay that control the pumps. The load current is then sampled for a set amount of time, and the average draw is calculated. Since the bilge pumps are inductive loads, the current they draw is higher if there is resistance on the motor shafts due to water in the system. That way potential leaks are detected. If the system suspects a leak, it will run the bilge pumps until the next leak check is scheduled. The remote operator will also be warned via level 2 and the 433 MHz radio. When the iridium subscription is enabled, that will provide a third reliable way of warning the operator. Since the iridium message will be sent to an email address, a system can easily be made to warn one or multiple people via SMS by using a service such as offered on www.IFTTT.com so that immediate efforts can be made to investigate the situation further. The leak warning is sent on the iridium device by calling the function named *report_leak()*.

7.11 Disabling of devices

Requirement *B.REQ.2.11.5*, disabling of device power, was implemented by dedicating a section in the polling loop to set dedicated I/O pins high or low in accordance with a set of variables stating desired power settings. The section utilizes simple conditional statements. As described in Chapter 6, some I/O pins go directly to the sleep inputs on the devices whilst others control relays. A function named *setPwrSettings* was implemented to meet requirement *B.REQ.2.03.4*, manual disabling of power for devices. The function takes a string as input argument. The string is parsed to disable the desired devices. The required input string is part of the NMEA0183 based message protocol designed for interfacing the operator via level 2, the 433 MHz radio and iridium.

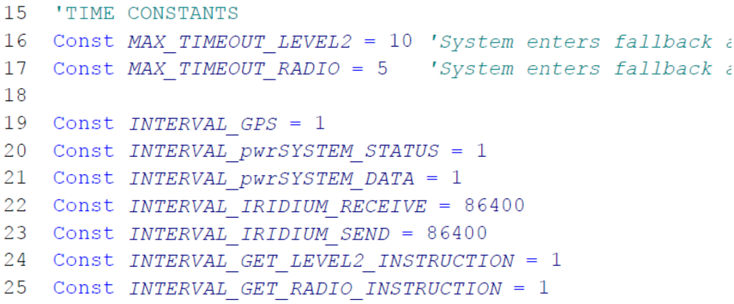
7.12 Autopilot

The PI controller was implemented as a function named *autopilot_piController* which takes desired COG as input argument. The function is called very early in the polling loop because that ensures that the integrated error value will be estimated exactly once every second. The PI controller is implemented with logic that prevents integrator wind up. Since the integrated error value is calculated at every time step and stored in a variable, this can easily be achieved by checking if the integrated error value leads to a control output that exceeds the allowed output range. If the calculated control output becomes saturated, the integrated error value is discarded and the integrated error from the previous time step is kept instead. If the control output has become saturated, but the error has the opposite sign as the control output, the integrator error value is updated. The result of the algorithm is that wind up of integrated error is prevented. The value of the integrator error in the event of control output saturation can be expressed mathematically as shown in equation 7.1.

$$\int e(t)dt = \frac{1}{K_i}(U_{saturated} - K_p e(t)) \tag{7.1}$$

7.13 Using the Program

Depending on the length of the operation, the operator should be familiar with some aspects of the implementation of the level 1 system. For normal use, the system is plug and play, but some behavioral changes can easily be changed by altering constants in the source code. This includes timeout settings for entering fallback mode, device polling intervals, initial system settings and leak detection interval. Figure 7.10 shows a section of the code where such settings can be changed. It was decided that the polling intervals should be constant during operation to avoid potentially hazardous situations caused by user error. For instance, if the interval for reading iridium messages was accidentally set to one year by the operator during a stressful event in the middle of an operation, that could lead to loss of the USV. Increased time intervals yields lower power consumption, frequent leak checks will increase power consumption significantly, and frequency of Iridium transmissions impacts the cost of operating the USV, otherwise, there are no particular considerations that should be taken.



```
'TIME CONSTANTS
Const MAX_TIMEOUT_LEVEL2 = 10 'System enters fallback a
Const MAX_TIMEOUT_RADIO = 5   'System enters fallback a

Const INTERVAL_GPS = 1
Const INTERVAL_pwrSYSTEM_STATUS = 1
Const INTERVAL_pwrSYSTEM_DATA = 1
Const INTERVAL_IRIDIUM_RECEIVE = 86400
Const INTERVAL_IRIDIUM_SEND = 86400
Const INTERVAL_GET_LEVEL2_INSTRUCTION = 1
Const INTERVAL_GET_RADIO_INSTRUCTION = 1
```

Figure 7.10: Definition of time intervals.

Chapter 8

Intergration and Testing

Testing and integration were performed at multiple levels. On a very low level, the different implemented functions were tested. That included testing of functions to control the devices that were interfaced with the CR6 computer in level 1. When it had been established that the program functions were working as intended when they were executed separately, the level 1 system could be tested in its entity. When the level 1 system was found to be functional without errors, it was tested with the level 2 system and the manual control software that had been developed in parallel to the level 1 system.

8.1 Testing of Functions

The program that was implemented on the CR6 had 30 user implemented functions. In order to test the functions, different input arguments were provided to see if unthought of results occurred. The sub-functions for error checking in messages were simply tested by creating a list of different messages, some who had the correct format and checksum and some which were erroneous and then comparing the results with the prepared solution. Although attempts were made at testing all functions thoroughly to identify bugs at an early stage, several mistakes were made. For instance, "TRUE" which is commonly defined as "1" in programming languages is defined as "-1" in CR Basic. This lead to bugs because the constant value "ERROR" had been set to "-1". Therefore the return values "ERROR" and "TRUE" yielded the same result and caused problems until "ERROR" was redefined to "-101".

8.2 Testing of Integrated Devices

The integrated devices were tested in different ways. For all the devices, the error detection system was tested. The table in Chapter 7.8 was gone through and it was ensured that the different errors were detected on occurrence. In addition, the devices had to be tested in accordance with their characteristics. The GPS was placed in near the window in the office and the information received was checked. The horizontal accuracy was within 2 meters and the unit typically had a fix on six to seven satellites. For the rockBlock+ Iridium device was tested by using a function that copies the content of the outbound buffer to the inbound buffer. That way the sending and receiving of a message could be simulated even without an activated iridium subscription. All the basic functions for the device were tested apart from the function that initiates a satellite communication session.

The BlueSolar device was not tested to full extent. The error detection on the communication link was tested. The functions for enabling and disabling the relay and measurement of load current was tested and compared to values measured on the power supply unit that simulated the batteries. However, a PV panel was not connected at this stage. The decision not to test the remaining BlueSolar functions such as obtaining PV panel power was made in order to save time in an attempt to stay on schedule. It was also assumed that the basic device functionality including battery charging and MPPT algorithm would work. The power consumption of the different devices was checked by using the built-in current measurement functionality in the power supply. GPS and Iridium devices were tested in standby mode and in normal operating mode. It was found that the rockBlock+ current was surprisingly high at startup at (300mA at 12 V) however, that did not require changes to be done. The leak detection system was tested by connecting a 12 DC motor to the system. The increased load from water in the pumps was simulated by increasing the resistance on the motor shaft.

8.3 Level 1 System Test

When all the integrated devices in level 1 had been tested separately, the programmed polling loop and state machine could be tested. The following devices were connected: GPS, Iridium device and BlueSolar MPPT Controller. The motor controller device was not interfaced, but the PWM control signals were shown on an oscilloscope. The bilge pumps were not connected, but one 12 V DC motor was connected to the load output of the BlueSolar device. Since the devices had not yet been mounted in the case with the relays, the wiring was done differently than in the finalized system, but the set up made it possible to test the main functions.

Since level 2 was still under development in parallel, an Arduino Uno used to simulate level 2. Arduino Uno is a single board microcontroller that is ideal for prototyping. It is cheap and has dedicated hardware for serial communication at RS-232 voltage levels (30). Pin 0 and 1 were used on the Arduino. Since the Arduino Uno requires a 5 V power input, one of the outputs on the power supply unit was set to 5 V and the grounds for the 5 V output and 12 V output on where connected. By doing so, the Arduino Uno and the CR6 had common ground which is required for the serial interface to work. The program for the Arduino was written using Arduino software version 1.8. The language used is very similar to Python. A simple polling loop was implemented so that the device would respond to the level 1 system in the event of incoming messages. An NMEA0183 checksum function had to be implemented on the Arduino to test that the checksum functionality on the level 1 system worked. With the Arduino Uno it was possible to test the system command sequence from level 2, through the CR6 and to the outputted PWM signal that was displayed on the oscilloscope. It was also possible to check that fallback mode was entered if the Arduino provided erroneous checksums or if the TX or RX wire was disconnected. A benefit of using the Arduino Uno is that it has built-in functionality for viewing the data stream on the serial ports via the USB cable to a windows computer. By combining this functionality with the public variable monitoring functionality in the PC400 software, it was relatively easy to get a thorough understanding of the system behaviour at all times.

For the connectors that were not connected to devices (see figure 8.2), it was checked that the connections were made to the right pins on the CR6 computer by using a multimeter, and the correct relays.

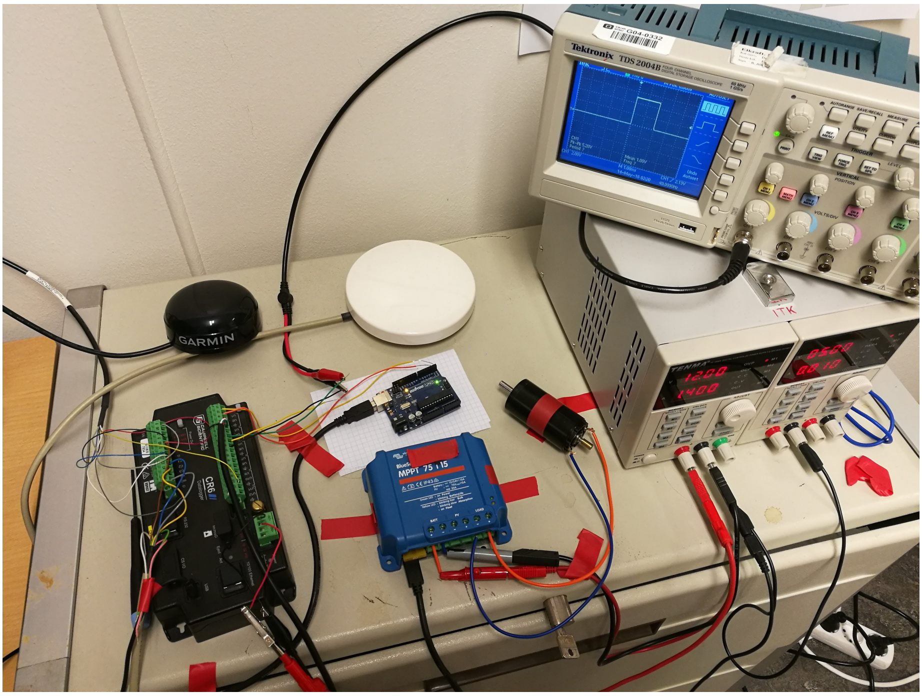


Figure 8.1: Level 1 initial test

As soon as the final adjustments had been made to the level 1 system and implementation was near complete, the equipment was mounted in the level 1 case by the workshop at the department. After the mounting of the devices in the case was complete, some of the I/O pins had to be changed to allow for nicer wiring. That was done through constants in the program. After completing the level 1 system was re-tested. The implemented error logging was enabled. No errors were detected apart from a few initial *"no_response"* errors caused by the startup time required for the GPS, Iridium device and BlueSolar unit. At this point the 433 MHz radio transceiver had arrived so it was included in the test set up.

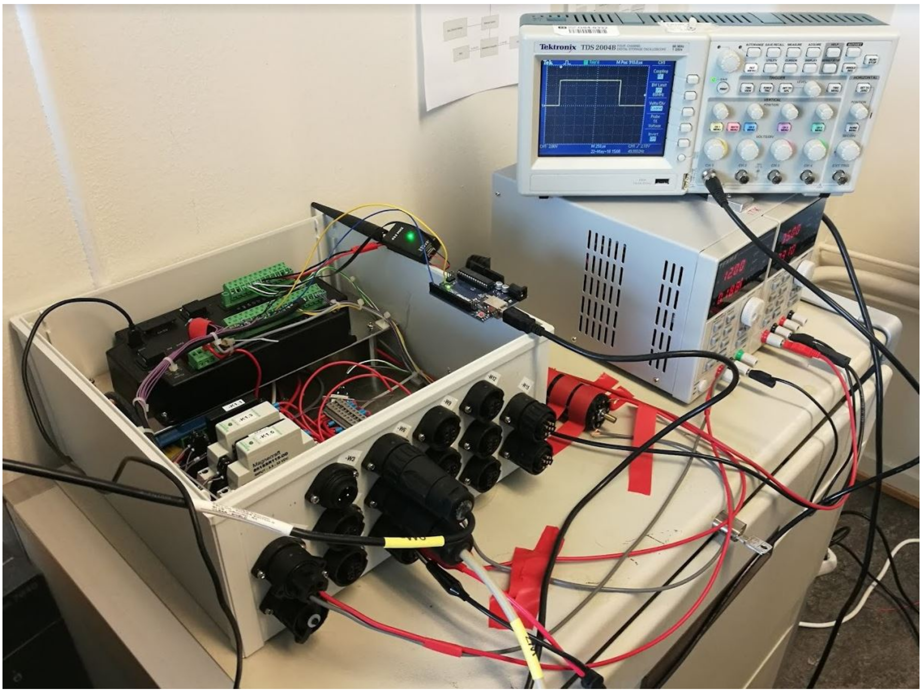


Figure 8.2: Level 1 stand alone test in IP67 case.

8.4 Integration with Manual Control Software

The Manual Control Software that was made by Artur Zolich was tested with level 1. All the functions in the Remote Control Interface Protocol (*B.REQ.2.07.1*) were tested, including setting the rudder angle, setting the thruster power and setting the relays. The automatic fallback behaviour was also tested and timed. The 433 MHz interface antenna unit can be seen next to the Arduino Uno in figure 8.2.

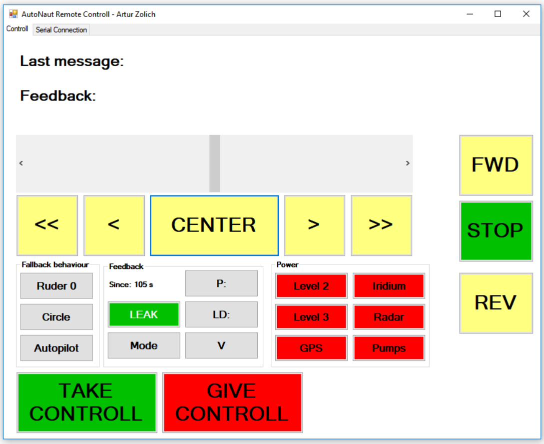


Figure 8.3: Screenshot of an early prototype version of the Manual Control SW for Windows made by Artur Zolich.

8.5 Integration with Level 2

The development of the level 1 system was coordinated with the development of the level 2 system. When both systems were ready, they were connected with a test cable. The test cable is identical to the cable that will be used in the finalized system, only shorter. The cable provides power for level 2 via a relay in level 1. Initially, the two cases that constitute the core of level 1 and level 2 where placed next to each other in the office. They were connected with the interfacing cable and level 1 was connected to a power supply instead of a battery. It is beneficial to use a power supply since the maximum output current can be controlled to limit or avoid the detrimental effects of short circuits.

Level 1 was left to operate without direct control input, but the internal system parameters were observed using a USB cable and the PC400 software public variable monitoring functionality. The level 2 system was interfaced wirelessly using the

Ubiquity Rocket the same way as it would later be interfaced during sea testing. When the subsystems were tested together, requirement *B.REQ.2.07.3*, level2 control mode, was tested. This was tested by ensuring that level 1 remained in the correct state as long as level 2 was operating normally, and that rudder and thruster instructions yielded the correct PWM signal on the oscilloscope. In addition, requirement *B.REQ.2.08.2*, level 2 failure monitoring was checked. Failures were tested in different ways. The communication cable was disconnected and reconnected to assess the behaviour, and the level 2 system was disabled to see that the fallback state was entered in level 1. The PWM output signal was also checked during these tests in addition to monitoring of internal error reports using the PC400 software's public variable monitoring capabilities.

Requirement *B.REQ.2.11.6*, disabling of device power, was also tested. It was found that level 2 failed to restart unless the reset button on the BeagleBone Black computer was pressed. Therefore some changes had to be made in the level 2 system. It was also found that there was a minor issue with the communication between level 1 and level 2. Messages sent from level 1 to level 2 seemed to fail the checksum test. Since the interface had been tested rigorously at an earlier stage by using the Arduino Uno with level 1, it was assumed that the transmitted messages from level 1 were correct. It was found that the probable cause for the problem was a combination of two things. The transmission line between level 1 and 2 was poorly protected against RFI since the test cable was unshielded and a bug in the software on the BeagleBone caused erroneous behaviour when the checksum calculation failed. The baud rate on the serial line was therefore reduced from 38400 to 9600 and the bug in the software in level 2 was solved.

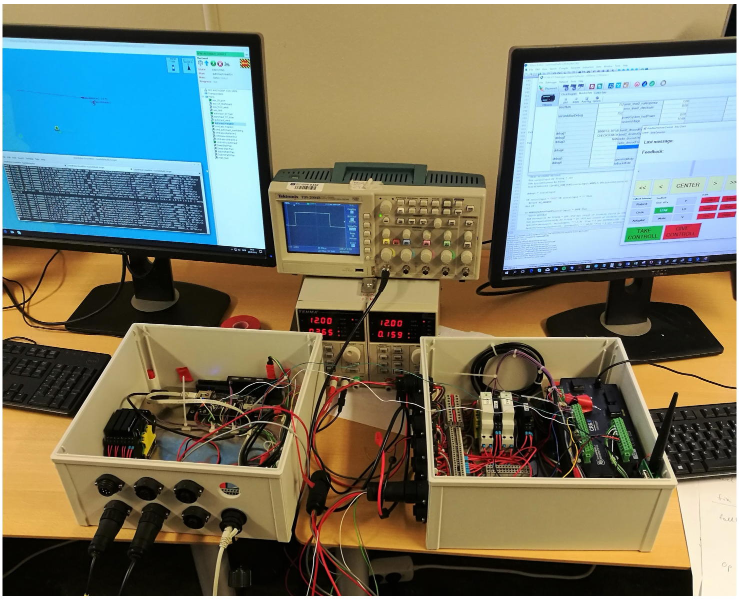


Figure 8.4: Level 1 and Level 2 Integration Test.

Chapter 9

Verification and Validation

System verification seeks to answer whether a system is built and functions according to the system requirements, while the process of validation answers whether the system meets the operational needs (6). A strict boundary was not set to separate between integration and testing and system verification in the project. Some of the tests described in Chapter 8 could be said to have verified system functions. However, since the subsystems were not set up exactly as they would be in the USV, it would have been unwise to jump to conclusions at that stage. For example, the actual motor controller was not connected, and a power supply unit was used instead of batteries.

Due to time constraints, the USV system was not validated to full extent, but the level 1 system was verified and most of the operational needs that are provided by level 1 were validated.

9.1 Verification

Prior to the sea trial, several system functions were verified at the car park at Gløshaugen. The aft section of the USV was attached to a trolley and one battery was connected. A control station was established at the entrance of building D. The following system requirements were verified: Load Power Monitoring (*B.REQ.2.11.2*), Disabling of Device

Power (*B.REG.2.11.5*), Device Error Monitoring (*B.REQ.2.08.1*), Level 2 Failure Monitoring (*B.REQ.2.08.2*), Bilge Pumps Control (*B.REQ.2.08.4*), Remote Control Interface Protocol (*B.REQ.2.03.1*), all requirements in the manual control category (*A.REQ.2.03*) and Output of position, COG and SOG (*B.REQ.2.06.4*). The remaining requirements could not be tested because of the following reasons: Testing of in-port charging unit (COTS) was not prioritized, no PV panels were fitted because it would require the other hull which would have required some extra hours for assembly, leak detection was not prioritized, the iridium subscription was not activated. Requirement *B.REQ.2.07.4* was tested to a certain extent by pushing the USV section in different directions and observing the rudder response. It was found that the rudder went to the port side when it should go to starboard. The testing of the other requirements listed is self-explanatory. Most of the allocated time was spent testing functionality related to level 2, which is more complex and therefore a lot more difficult to verify at a car park.

Figure 9.1: System verification at the Gløshaugen car park.

9.2 Field Test at Trondheim Biological Station

Trondheim Biological Station (TBS) is located next to the Trondheim Fjord approximately 3 Km North-West of Trondheim City Centre. Two days were scheduled for assembly of the USV and sea trials. The field testing took place on the 28th and 29th of May 2018. Transportation of equipment from the facilities at Gløshaugen to TBS and assembly was time-consuming, so no system tests were done during day one. However the USV was lowered into the water for 15 minutes to ensure that the seals were watertight.

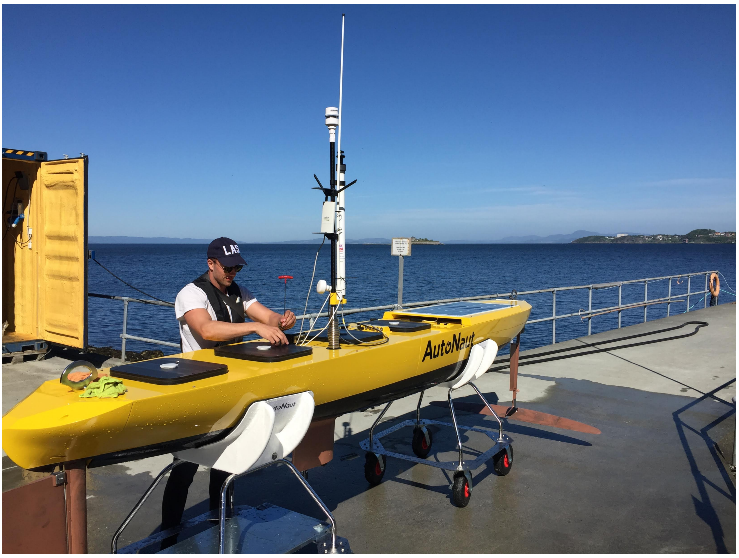


Figure 9.2: Preparing the USV for sea trial at TBS. Photo: Sølve D. Sæter

At 10:30 on May 29th the USV was lifted into the water from the dock at TBS. A control station operated by Sølve D. Sæter was set up on the dock. The Manual Control Software that enabled communication with the level 1 over the 433 MHz link was taken on a PC together with a radio transmitter on the support boat. It was found that the range of the 433 MHz link was limited to approximately 30 to 50 meters depending on the angle. The reason for the limited range was that the antenna was fitted inside the IP67 case as a temporary solution. The Ubiquity M5 on the other hand, had a range of 400 meters from the dock when in line of sight. The support boat used was Fjøset II, which is an 18 feet Polarcirkle with a Yamaha F150 motor. The boat is well suited for the support role because it is stable and has a low railing. Communication between the operator station at the dock and the support boat was established over the telephone. Both at the operator station and in the support boat, VHF radio was available for contacting other ships and boats in case of emergency situations.

During the planning phase of the sea trial, a document was prepared with a prioritized set of activities. In addition to verifying and validating the system, maneuvering tests were scheduled to provide future system developers with a data set for tuning purposes. To verify and validate the level 1 system thesis, several tests were performed and gathered data was analyzed.

Figure 9.3: USV and Fjøset II, Munkholmen in the background. Photo: Sølve Sæter

Requirement	ID	Comment
Load power monitoring	B.REQ.2.11.2	No errors detected
PV panel power monitoring	B.REQ.2.11.3	No errors detected
Remaining Energy Estimation	B.REQ.2.11.4	Not implemented
Disabling of device power	B.REQ.2.11.5	Not tested to full extent
In-port charging	B.REQ.2.11.6	Not tested
Device Error Monitoring	B.REQ.2.08.1	Worked. Should be configured.
Level 2 Failure Monitoring	B.REQ.2.08.2	Tested extensively
Leak detection	B.REQ.2.08.3	Test failed. Requires tuning.
Bilge pumps control	B.REQ.2.08.4	Worked. Only tested 1/3 pumps.
Remote Control Interface Protocol	B.REQ.2.07.1	No errors detected.
Manual Control Mode	B.REQ.2.07.2	No errors detected.
Level 2 Control Mode	B.REQ.2.07.3	Level 1 system responds as intended.
Fallback Autopilot Mode	B.REQ.2.07.4	Worked. PI Controller can be tuned better.
Remote Control Interface Protocol	B.REQ.2.03.1	No errors detected.
Rudder Angle Control	B.REQ.2.03.2	No errors detected.
Thruster Control	B.REQ.2.03.3	No errors detected.
Disabling of Power for Devices	B.REQ.2.03.4	Not tested to full extent.
Iridium Communication Link	B.REQ.2.06.1	Not tested during sea trial.
Radio Communication Link	B.REQ.2.06.2	Limited range due to antenna mounting.
Output System Energy Parameters	B.REQ.2.06.3	No errors detected.
Output Position, COG and SOG	B.REQ.2.06.4	No errors detected.
Output Leak and Error Status	B.REQ.2.06.5	Not fully implemented.

The sea trial consisted of the following tests:

1. Leak Test
2. Manual Control - Rudder
3. Manual Control - Thruster
4. Manual Control - Enable / Disable Level 2
5. Manual Control, Maneuvering Test 1 - Full thrust, 20^0 rudder port.
6. Manual Control, Maneuvering Test 2 - Full thrust, 45^0 rudder port.
7. Manual Control, Maneuvering Test 3 - Zero thrust, 20^0 rudder port.
8. Manual Control, Maneuvering Test 4 - Zero thrust, 45^0 rudder port.
9. Level 2 Active, Waypoint Mode
10. When in NORMAL mode, go to MANUAL mode
11. When in MANUAL mode, go to NORMAL mode
12. When in NORMAL mode, deactivate level 2 and go to FALLBACK
13. Fallback Autopilot, zero thrust
14. Fallback Autopilot, maximum thrust

A lot of time was also spent testing the level 2 navigation system. Since level 2 relies on level 1, these tests also tested level 1 indirectly. The level 1 system was monitored during the level 2 tests. Many of the level 1 requirements were verified as part of the tests listed. For instance, PV panel power monitoring, which is a requirement, did not require a particular test, but the data stream was assessed during testing of other functions. For every test that was performed a written log was kept with time stamps to ease the data analysis.

Test 1, Leak Test was performed on the dock. The leak check interval was reduced to 60 seconds and the hose connected to bilge pump 2 was put in a bottle containing

Figure 9.4: Controlling the USV from Fjøset II by radio link. Photo: A. Zolich

fresh water. The system did not report leak because the current increase at pump activation was lower than the threshold value. This happened because only one out of the total of three bilge pumps were connected. Therefore the measured power increase at activation was less than the expected consumption for a no-leak situation. The remaining two bilge were not connected because the level 1 box was placed in the aft compartment in the USV and not in the forward compartment as planned for future operations. Since the level 1 case was placed in the aft compartment, it was not possible to connect all bilge pumps due to lengths of cables. The reason the case was placed in the aft compartment was to limit the number of cables that had to be fitted through the glands between the watertight hulls. The leak testing must be repeated in the future when all bilge pumps are connected. The threshold value can easily be changed in the CR6 program.

Test 2, 3 and 4. (manual control). These tests were performed prior to leaving the dock to ensure that the USV responded to manual control inputs. The level 1 system send status messages to the Manual Control Software which contain power enable/disable status, but test 4 could also be verified by observing that the control station on the dock lost contact with level 2 when the level 2 subsystem was disabled by level 1.

No irregularities occurred during the maneuvering tests. Data was recorded and will allow for future system developers to make a better model for the USV dynamics. For each maneuvering test the USV performed a full circle. This was done because it will make it possible to better understand the impact waves and wind has on the USV. If only a short rudder step response test was performed, the tests would have to be repeated numerous times for obtaining useful data. That was not prioritized due to project time constraints.

Test 9 was the formal test that could validate an operational need. It was found that the advanced navigation system in level 2 requires more tuning, but the interface protocol between level 1 and level 2 worked fine. Due to CPU overload in the BeagleBone Black computer in level 2 during an attempt at simulating a collision avoidance scenario while, the level 2 system failed to answer messages from level 1 and the system entered fallback mode. This was registered in the internal log and messages where sent to the manual control software via the 433 MHz radio link. Test 12 was intended for simulating a situation where fallback was necessary, by deactivating the level 2 system, but it shows that the level 1 behaved as intended in an unforeseen situation.

9.3 Power Monitoring

The power monitoring functionality was tested for the entire sea trial. Figure 9.5 shows power system data from a 10 minute period of the sea trial. A few things should be noted. Firstly, only one PV panel was connected to the power system and secondly, the thruster's power consumption is not measured, but must be estimated when activated. It was sunny during the 10 minutes.

In the first 200 seconds, advanced waypoint navigation was active and the rudder was constantly adjusting. After 200 seconds, the rudder was set to idle, so the seen current consumption is from the level 1 and level 2 systems. After 300 seconds, level 2 is disabled for a few seconds, it is then enabled again. Note that the consumption of level 1 alone is around 10 watts. The full data set also confirms this. After 370 seconds the fallback autopilot becomes active and it can be seen that the power consumption increases due to this.

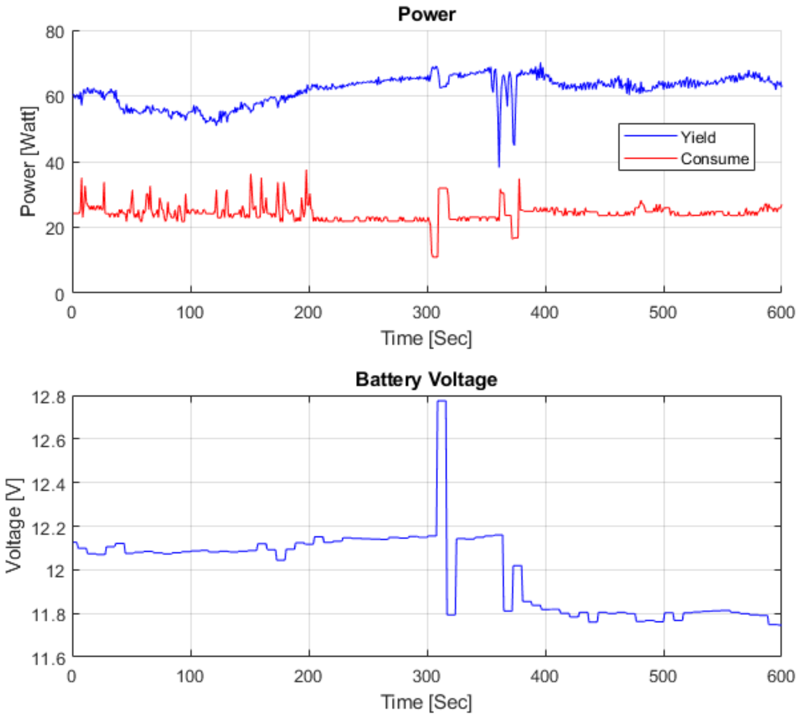


Figure 9.5: Matlab plot of power system data

9.4 Autopilot

The autopilot was tested twice. One test was performed with the thruster disabled so that the USV was moving at a slow speed. At the time when the autopilot was activated, the USV was moving in easterly direction. When the autopilot was activated, the USV redirected its course south. During the test, the USV was followed by the support boat. Based on observation, the USV stayed on course south, without visible oscillations. Unfortunately, the data from the 20-minute test run is unavailable.

The second autopilot test was performed downwind with the thruster at maximum to achieve a speed as high as possible. Higher speed in water leads to increased rudder response (faster system dynamics). The autopilot was tuned conservatively to avoid oscillations even in a worst-case scenario. As seen in figure 9.6, the USV oscillated around the desired course angle. This is because the model that was used for tuning the controller gains in chapter 6 does not accurately fit the real system. The plot shows that the autopilot worked in the worst case scenario that was created, but in severe conditions it could potentially become unstable. An improved model could be made based on data from the maneuvering tests so that the parameters could be tuned better. For a more optimal autopilot design, different sets of gain parameters could be used depending on sea state.

Note that the anti wind-up was working. As soon as the desired COG was reached, the rudder angle started decreasing. Observing the plot of the rudder angle, it would seem as if there is a constraint on rudder rate. However, that is not the case. The constant rudder turn rate is due to the low integrator gain. This was not detected during simulations prior to sea trials and shows that the autopilot was poorly tuned, still, it worked since it was tuned conservatively.

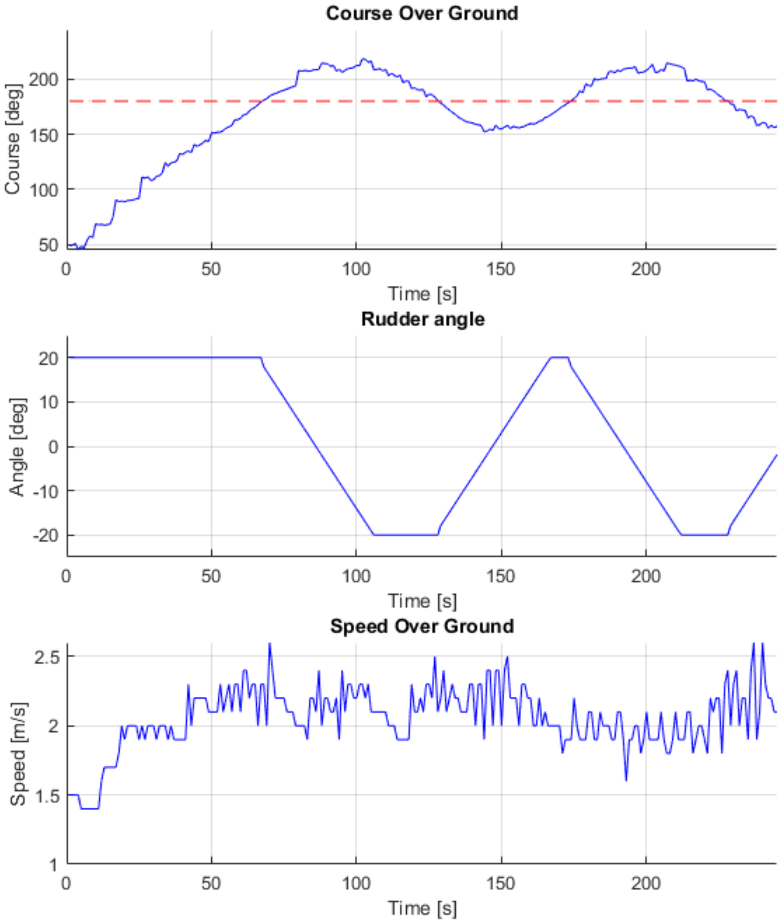


Figure 9.6: Autopilot test at maximum speed

Chapter 10

Conclusions and Future Work

10.1 Conclusions

First of all, it is essential to state that it is too early make firm conclusions regarding the designed and implemented system. Whether or not the implemented functionality proves valuable to future system users and can facilitate operational needs in a good way is too early to say. Further, the robustness of the system has not been tested very rigorously, only evaluated on a theoretical basis. On the other hand, most of the functionality that was implemented has been proven to work in a real sea trial, and even the unforeseen event of an accidental CPU overload in the level 2 system was handled automatically by graceful system degradation to the fallback state.

The behavioral design that was chosen for level 1 seemed to be effective and yet so simple that it was easily understood and efficiently used when tested during sea trials. Still, it should be pointed out that the operators during the sea trial were already familiar with the USV. It remains to see how future operators will understand and handle the system.

The fallback autopilot that was designed did work, but at relatively high speeds (for the USV), oscillations occurred. All through the controller worked well at wave propulsion; it is assumed that the system could become unstable in severe sea states.

Therefore the fallback autopilot should be tuned more before extended deployments.

The CR6 unit that was used for implementing the system behavior was a natural choice for its proven robustness. However, the supported programming language is limited in functionality, and debugging functions are very limited, that increases implementation time significantly. After getting familiar with the device over some time, a certain piece of advice would be to recommend the unit for applications that require high levels of robustness but to seek other alternatives if complex behavior is sought to be implemented.

During integration and testing, the perceived experience was that great care was taken to test everything. However, when functions were integrated and system complexity increased, it was found that it could have been beneficial to test the sub-functions to even greater detail. Without rigorous testing at early stages of development, the sea trial would have been impossible.

The system power consumption was sought to be kept low, and the sea trial showed that the level 1 fallback functions could be maintained with a consumption of 10 watts. With level 2 active, the power consumption is roughly double. It was observed that the active use of rudder does increase the consumption significantly, that should be taken into account when developing the advanced navigation system further.

10.2 Future Work

Several of the planned level 1 functions were not fully implemented. This applies to the Iridium system and the remaining power estimation system. The Iridium system was tested during system integration but was not verified at higher levels because it was chosen to delay the activation of the subscription. With the implemented functions it should not take long to get the system up and running. Estimation of remaining power, however, is a more complex task. A suggestion is to use the data for battery voltage in combination with real-time power yield and consumption. By assessing how the battery voltage fluctuates throughout a 24 hour period with different loads and charge state the remaining energy can probably be estimated. A more real-time estimation could be provided by using the real-time power data.

Another suggestion for future work is to analyze safety from a more external perspective. In this thesis, the safety focus was on the internal aspects of the system. The operator's role in the automated system should be discussed, and hazardous situations and how to avoid them should be analyzed.

Lastly, it is important to state that the implemented functions provide a relatively low degree of autonomy. However, they do facilitate full autonomy. That will require implementation of decision-making entities. For instance, the internal state error monitoring and system energy levels could be relied on for automatic decisions. All the data that is gathered in level 1 can be transmitted to the relatively powerful level 2 computer using functions that are already implemented. Functions are in order so that future developers can take the system to the next level.

References

[1] OECD, *The Ocean Economy in 2030.* 2016.

[2] Norwegian Ministry of Trade, Industry and Fisheries & Ministry of Petrolium and Energy, *Regjeringens Havstrategi*, 2016.

[3] AutoNaut.com, *Applications [online]*, 2018.

[4] U. Skoglund, *Bygger småsatelitter for å overvåke havet [online].* Gemini.no.

[5] NTNU, *AMOS Annual Report 2013.* 2013.

[6] C. S. Wasson, "System engineering analysis, design, and development: Concepts, principles, and practices," 2015.

[7] DNV GL, *DNVGL-OS-D203 Integrated Software Dependent Systems*, 2015.

[8] B. O. Agdal, "Green usv," 2017.

[9] D. Jones, "Sending singals to submarines," *New Scientist*, vol. 1463, pp. 37–41, 1985.

[10] Iridium Communications Inc., *Iridium 9602 SBD Tranceiver Datasheet*, 5 2015.

[11] Ubiquiti Networks, *Rocket M Datasheet*, 2015.

[12] Telenor ASA, *Telenor Dekningskart [online]*, 2018.

[13] Directorate of Fisheries, *Fiskeridirektoratets kart for akvakultur [online]*, 2018.

[14] Mediehuset Tek, *Norges største test av mobildekning [online].*

[15] N. Leveson, *Safeware : System Safety and Computers.* Addison Wesley, 1995.

[16] NASA, *FMECA PD-AP-1307.*

[17] NASA, *Apollo Program: Procedure for FMECA RA-006-013-1A*, 08 1966.

[18] A. Bouabidi et al., "Cycling performances and failure modes for agm and standard flooded lead acid batteries under partial state of charge mode," vol. 142, p. 04015027, 07 2015.

[19] K. Mertens, *Photovoltaics : fundamentals, technology and practice.* John Wiley & Sons, Ltd, 2014.

[20] Solbian, *Shading Technical Report*, 2014.

[21] T. I. Fossen, *Handbook of Marine Craft Hydrodynamics and Motion Control.* John Wiley & Sons, Ltd, 2011.

[22] CampbellSci.com, *Greenland Ecology [online]*, 2018.

[23] CampbellSci.com, *Costa Rica: Buoy System [online]*, 2018.

[24] CampbellSci.com, *Cloud Base and Visibility Sensors for Military Aviation [online]*, 2018.

[25] Campbell Scientific, *CR6 Measurement and Control System Operators Manual*, 2015.

[26] J. G. Kemeny et al., *Basic: a manual for BASIC, the elementary algebraic language designed for use with the Dartmouth Time Sharing System.* Dartmouth College Computation Center, 1964.

[27] Garmin, *GPS 16x Technical Specifications*, 2008.

[28] Victron Energy, *Data communication with Victron Energy products*, 2017.

[29] R. Williams, *Real-Time Systems Development.* Elsevier, 2006.

[30] Arduino, *Arduino Uno Datasheet.*