Concept-Design of a hydrofoil control system of solar boat

Final report

(Dottoresso, 2008)

Hui Du

Student number: 69940

Version: 6.0

Epomat

HZ University of Applied Sciences

Thesis project (CU08813)

Date: 19-06-2016

Title: Concept-Design of a hydrofoil control system of solar boat

Subtitle: Final report

Author: Hui Du

E-mail: du0002@hz.nl

School Name: HZ University of Applied Science

School Address: Edisonweg 4

Postcode: 4382 NW Vlissingen

School E-mail: info@hz.nl

Company Name: Epomat

Company Address: Apolloweg 24b

Postcode: 8938 AT Leeuwarden

E-mail: m.colon@epomat.com

Project Type: Thesis project

In-company mentor: lr. M. Colon

1st Examiner: Ir. J.P.C. Wubben

2nd Examiner: Ir.J.C.W. Haak MTD

Leeuwarden, June 19th 2016

Summary

The company Epomat and NHL Hogeschool are currently working on developing a new solar boat. This new boat will sail in the Top Class competition called "Dutch solar challenge", a ranking dominated by highly specialized firms and universities. In order to get a better place in the competition, this solar boat needs to be equipped with a new technology for decreasing drag called hydrofoil.

What are hydrofoils? The hydrofoil technology is capable of making the boat run faster. A hydrofoil is a wing placed underneath a boat to lift the hull out of the water. As speed gains, the hydrofoils lift the hull of the boat out of the water, decreasing friction in the water and thus allowing higher speeds. There are two types of hydrofoils: a self-stabilizing water piercing configuration and a fully submerged configuration that requires a controller.

The solar boat to be developed is designed with a fully submerged configuration hydrofoil; therefore, a controller is necessary. The researcher is responsible for designing the control system and writing the software of the control system. The whole system is designed as an automatic control system via using PID controller. The researcher designs the action of each tuning value of the PID controller. Different tuning value should be designed to effect on the height, pitch and roll value of the boat separately. Different sensors to detect the position of the boat are applied. The data got from the sensors to the desired point needs to compare. Correcting the difference via the controller to lift the hull out of water, keeps the boat stable and maintains it at the constant height.

The report is divided into 6 chapters: Introduction, Theoretical framework, Method, Result, Discuss and Conclusion & Recommendations. The product is designed under the design method of V-model. System design, subsystem design, component design and detailed design of the control system are included. The objective of the report is to provide a clear description of the execution of the project and detailed design of the hydrofoil control system.

As result of the research and regarded as an added value for the company Epomat as well for NHL Hogeschool, are the facts that the accuracy of the sensors was improved and the controller system was designed in particular. Based on the test result of the sensors and the test model, the software to handle the data got from the sensors was written and optimized until the test model can work and meet the requirements. The whole system was divided into two major parts, stability and height control subsystem. A test model was built for testing the stability subsystem and after several testruns with the test model it was proved that this subsystem meets the requirements. The other height subsystem was designed with a test plan.

Due to the lack of time and that the hydrofoil was not finished in building, the whole system was not tested in practice yet. For further research it is recommended, the whole system should be tested by more realistic sea trials in order to optimize the whole system design.

Foreword

The researcher Hui Du, majored in mechatronics, is a graduating student from HZ University of Applied Sciences, who worked for Epomat as an intern to finish the thesis project.

This thesis project is to design a hydrofoil control system which can make the boat run faster.

This report only has the purpose of studying; any commercial use of the report without permission of Epomat is prohibited.

In the process of the project, the researcher received a lot of assistance of many people within NHL Hogeschool and Epomat and is very grateful to them.

A special acknowledgment to Mathieu Colon, the manager of company Epomat and the in-company mentor of the researcher, he helped the researcher a lot to finish the project.

Hui Du
Leeuwarden, June 19th 2016

List of abbreviations and technical term

Hydrofoil: There are two meanings of hydrofoil nowadays. A hydrofoil is a wing can be controller or a fixed lifting surface, which operate in the water. Boats that use hydrofoil technology are also simply termed hydrofoils. To avoid misunderstanding, in this document, the hydrofoil only means the lifting surface or wing that operates in water, which are similar to airfoils of airplanes in appearance and purpose.

IMU sensor: Inertial measurement unit. Another name of the IMU sensor is compass sensor.

PCB: Printed Circuit Board.

PWM: Pulse width modulation. In Arduino, the range of PWM value is 0~255. By adjusting the value of PWM, the velocity of the motor can be controlled.

PID controller: The letter PID means proportional, integral and differential action separately. The PID controller is the most common controller. A PID controller calculates the error value as the difference between a measured [Input] and a desired set point. This controller contributes to minimize the error by adjusting [an Output].

Kp: The value of proportion in PID controller. In this document, Kp1, Kp2, Kp3… all means proportion, but with a different numerical value.

Ki: The value of integral in PID controller. In this document, Ki1, Ki2, Ki3… all means integral, but with a different numerical value.

Kd: The value of differential in PID controller. In this document, Kd1, Kd2, Kd3… all means differential, but with a different numerical value.

Tuning values: The collective name for Kp, Ki and Kd. Adjusting these values will change the way the output is adjusted. What are the "right" tuning values to use? There isn't one right answer. The values that work for one application may not work for another, therefore the tuning values need to be adjusted in practical life.

C_L: Coefficient of lift.

AOA: Angle of attack (measured in degree).

A: Surface area of foil (measured in m^2).

V: Velocity (measured in meter per second (m/s)).

ρ: Density of water (measured in kg/m^3).

Table of Contents

Tables of figures

Table of tables

Chapter 1 Introduction

In this chapter, the research assignment is introduced and described. Including the background information of the company and project, the complete description of the assignment, problem definition related to this assignment. Moreover, the report structure is described in this chapter.

1.1 Company Introduction

Epomat was founded in 2010. The detailed and contact information of the company Epomat was listed in table 1.1. The company Epomat not only has the background of the maritime industry, but also the desire to innovate and to find solutions, a passion for the product and alternatives in composite for existing products and problems. Figure 1.1 demonstrates an overview of company projects. Epomat is one of the few composite production companies in the Friesland area who dedicate themselves to all-round composite production. Epomat has all the knowledge and tools available to design and produce products from beginning to end. For more than 5 years, Epomat has been active in the wind industry, providing composite maintenance, repairs and solutions for the wind turbine blades. They are certified for working at heights based on GWO standards, hold an IRATA level 1 certificate and are in possession of a VCA certificate. (Epomat, 2015)
Having a background in the maritime/Yachting and wind industry, Epomat is specialized in solar races. One of the objectives of Epomat is to design a new solar boat which can win a place in the Top class competition. In order to reach that goal, the company Epomat and NHL Hogeschool cooperated with each other to optimize the solar boat.

Company name:	Epomat
Website:	info@epomat.com
Street address:	Apolloweg 24b
Post code:	8938 AT Leeuwarden
In-company mentor:	Ir. M. Colon
E-mail:	m.colon@epomat.com
Phone number:	0646894405
Level of education:	MSc

Table 1. 1 Company information

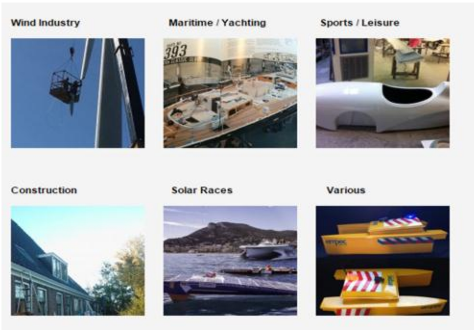


Figure 1. 1 An overview of company projects

1.2 Problem Exploration

This phase contains the background information and description of the assignment. Additionally, the problem definition, main research question and sub questions are described as well.

1.2.1 Assignment Background

The company Epomat and NHL Hogeschool are currently working on developing a new solar boat. Figure 1.2 shows an example. The new boat will take part in the Top Class competition, a ranking dominated by highly specialized firms and universities.

There is a big solar boat team for building and designing the new boat. This team consists of the employees from Epomat, teachers and students from NHL Hogeschool. Everyone have different assignments like designing the electrical circuit of the solar boat, designing the solar panels, determining which motor will be used and so on. One of the assignments is to design the control system of the hydrofoil.

Almost all the boats in the top 3 place at Top Class competitions are built with hydrofoils. Hydrofoils are wings placed underneath a boat. Hydrofoils generate lift to lift the hull out of the water, which reduce the resistance in the water and save energy. In order to make the boat run faster, the hydrofoil technology is essential for the new solar boat.

Figure 1. 2 The configuration of the hydrofoil to be developed

1.2.2 Assignment Description

The company Epomat is regarded as the client who has formulated the following requirements:

The main target of the assignment is to design a hydrofoil control system to make the boat run faster while keeping stable. According to previous research within NHL, the types of sensors and microcontrollers were determined. Therefore, now the task it to design a hydrofoil control system with the determined sensors and microcontroller. The design contains the electric circuit of hardware and algorithm of software of the control system. The controllers need to be designed more in detail and the software must be designed and written. Additionally, the accuracy of the sensors needs to be improved yet.

1.3 Project Background

What are hydrofoils? Hydrofoils are wings placed underneath a boat to lift the hull out of the water, which operates in the water. As speed gains, hydrofoils lift the hull out of the water, decreasing friction in the water and thus allowing greater speeds. There are two types of hydrofoils: self-stabilizing water piercing configuration and fully submerged configuration that requires a controller.

In the project, the mechanical part of the hydrofoil has been designed already. The real appearance of the hydrofoils is determined. (Shown in fig 1.2) The front wing (Shown in fig 1.3) is configured in such a way that it is actually three wings instead of one: the one in the middle and two side wings. (Kagendijk, 2015) The rear wing is designed as one piece.

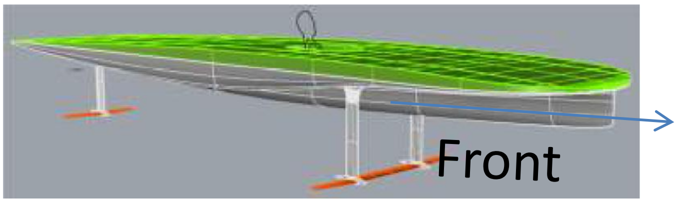


Figure 1. 3 Four wings of the hydrofoil

Via steering the two side wings, the stability of the boat can be ensured. Additionally, the middle wing at front and the sole rear wing should be adjusted to control the height of the boat. The height means the distance between the bottom of the hull and the water surface. As a result of making sure the height and stability, the hydrofoil can ensure that the boat sails out of the water straight and level.

1.4 Problem Definition

The hydrofoil control system contains a height control system and a stability control system. The stability control system must keep the boat stable when it is sailing, while the height control system is maintaining the stable height between the bottom of the hull and the level of the water. The height adjusting and the stability are seen as two contradiction factors. How to control the hydrofoil under any condition is the challenge to be met and is regarded as the problem.

1.5 Problem Analysis

The problem to be solved is incorporated within a long running project. The previous research determined which two kinds of sensors would be used. One was ultrasonic sensor. It can get the information of the height. The other one was IMU sensor, which can detect the roll and pitch value. Figure 1.4 demonstrates the position of these two directions. These two values contributed to determine the position of the boat. These two sensors contributed to detect the height and determine the stability of the boat. A PCB was built with these two sensors and assembled with Arduino board. However this PCB was a proof of previous concept, not a final product.

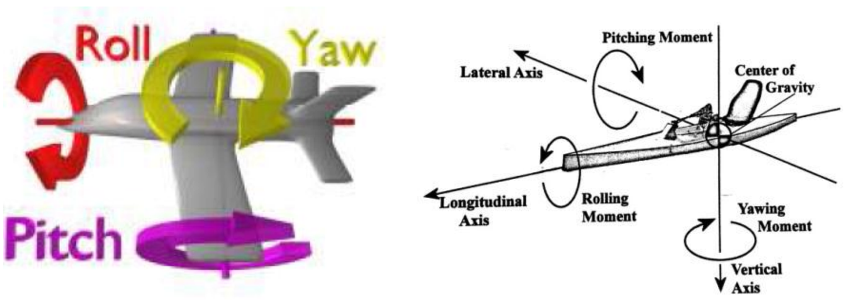


Figure 1. 4 Aeronautical direction definitions

The signals of the sensors were determined as the input to the Arduino board, but the waves in the water lead to inaccurate height value got from the ultrasonic sensor and the movement of the boat leads to inaccurate roll and pitch value got from the IMU sensor. Therefore, the accuracy of the input signal needs to be improved. What need to be done next is that based on the signals from the sensors, the output of the Arduino board controls the actuators. The main challenge is to make a design of the hydrofoil control system and program this controller to the Arduino board, which can operate in the harsh conditions.
Now that the type of sensors is known, it is time to make a test of the printed circuit board (PCB). The choice was made to use the Arduino platform and make a test of PCB. Eventually a final PCB was made, which was also applied an Arduino shield. (Veen, 2015)
Above is the reference from the report of previous research.

1.6 Research Objectives

In order to solve the problem a research was conducted with the following objectives:

1. To understand the working principles of hydrofoil.
2. To improve the accuracy of the sensors.
   The sensors are utilized to detect the position of the boat. Only when the sensors are accurate enough, the control system can work. Each software of getting the data from the different sensor needs to be optimized by testing and analyzing the result of the tests.
3. To develop a control system to keep the boat horizontally stable when it is sailing.
   Keeping the boat horizontally stable means that keeping it in a steady and safe position. Based on the aeronautical theory (Shown in fig 1.4), it is essential to keep the roll value and pitch value moving in range to avoid severe accidents. By adjusting the hydrofoil, the direction in roll value and pitch value of the boat can be changed.
4. To develop a second control system that maintains the boat at constant height.
   Maintain the boat at constant height means that maintaining the distance between the bottom of the hull and the water surface. By adjusting the angle of attack (AOA) of the middle wing in the front and the rear wing, the height can be changed. In the other words, adjust the front and rear wings of the hydrofoil, to keep the boat verticality stable.
5. To write the code of the controller.
   Not only the controller should be designed, but also the code of the controller should be programmed in the software based on the design.

6. To combine both control systems into one main control system.
7. To verify the design by making a test plan and writing a user manual.
8. To verify the design by building a test model and test the design with it.

1.7 Main Research Question

Derived from the research objective the main research question could be defined as follows:
"What is the best cost-effective design of the control system of a hydrofoil that can maximize the speed of the boat while keeping the boat stable and safe?"

1.8 Sub Questions

In order to be able to answer the main research question, the research is split up into three parts.

1. Gathering basic **theoretical** information about the project, resulting in theoretical questions.[T]
2. Exploring the current position by raising **empirical** questions. [E]
3. Finding an improvement by raising **analytical** questions. [A]

Finally, the following sub questions must be answered:
T1. How to make the boat run faster while keeping the boat stable?
T2. How to waterproof the sensors without affecting the signals?
T3. Which method should be used to control the stability of the boat automatically?
E1. What should be controlled by the pilot manually?
E2. What else should be considered to keep the pilot safe under any conditions?
A1. How to reduce the error of the height measurement because of the waves?
A2. How to optimize the accuracy of the data got from the sensors?
A3. What is the range of the roll and pitch value of keeping the boat stable?
A4. How to check and adjust the angle of attack (AOA) continuously to keep the solar boat stable when the hydrofoil is working?
A5. How to check and adjust the AOA continuously to maintain the constant height when the hydrofoil is working？

1.9 Research Boundaries

In this research project, the researcher will design the controller of the hydrofoil control system. This controller system is only suitable for this hydrofoil. (Shown in fig 1.3) This hydrofoil will be finished in building next year so that the test in practice is not possible. The researcher has found a way to build a test model to prove that the theory of the designed controller and coding can work. On top of that, the accuracy of the sensors must be proved so that it is able to work in the hydrofoil working environment.

1.10 Report Structure

The report is divided into 6 aspects:

1. Introduction

Introduction contains background information about the project and defines the research problem.

2. Theoretical Framework

Theoretical framework contains knowledge, technique and theories related to the research problem, including the result of previous research.

3. Method

Method contains the design method and research method applied to the project.

4. Result

Result includes the important results during researching period. The layout of the result was determined by the design method.

5. Discussion

In this chapter, what the researcher has learned was described.

6. Conclusion and Recommendations

In this chapter, the conclusion to main research question and sub questions were described, including the recommendation for the further development.

Chapter 2 Theoretical Framework

In this chapter, previous research results and theoretical description are elaborated in detail, including the result of literature search.

2.1 Previous Research Results

Figure 2. 1 Existed PCB

The choice was made to use the height of the boat as input for the height controller. The height measurement is used for adjusting the angle of attack (AOA) of front and rear wings. Additionally, roll was chosen as the variable for the second control system used for stability. This system controls the side wings of the hydrofoil. Research was done for different sensors to measure height, pitch and roll. (Veen, 2015)

Above is the reference from previous research, the rough concept of the control system is to use the roll value as input to control the two side wings for stability and use the height and pitch value as input to control the front and rear wings for maintaining the whole hull at the constant height. The types of sensors were determined, and the previous researcher who also built a PCB (Shown in fig 2.1) with the ultrasonic sensor and IMU sensor, which proved that the value of height and roll can act as input value. When the hydrofoil is working, the ideal situation of the boat is that both the roll and pitch value are at 0 degree and the height maintains at 30 centimeter (cm). This PCB was assembled with Arduino board already. This PCB is only a proof of the previous concept. It is not feasible in practice because the quality is insufficient. It does not work every time and it is not waterproof. Besides, the accuracy of the input value is still an existing problem.

2.2 Theoretical Descriptions

This section shows literature search results. The literature was chosen according to the project requirements. To design and program a control system, it is essential to understand the working principle of control system, hydrofoil and microcontroller.

2.2.1 Hydrofoil Dynamics

The lift formula for hydrofoils is similar to aerodynamics: Lift = $(1/2)*\rho*A*CL*V^2$
A is dependent upon the actual wing shape. V represents velocity. Meanwhile, ρ means the density of the water. CL is coefficient of lift; this is a dimensionless number found in tables of

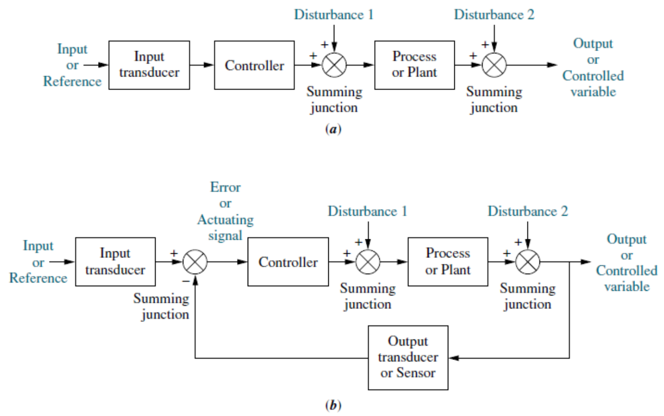
wing profiles as in theory of wing sections. (Vellinga, 2009)
Lift comes from the dynamics of the fluid in the area surrounding the foil. However, the lift can be enhanced by positioning the hydrofoil at an angle. This angle is called the angle of attack (AOA). (Radio corpotation of America, 1963)
As a result, changing the AOA contributes to changing the value of CL in the formula, which can change the lift.

2.2.2 How the Hydrofoil Works

When the boat is at low speed, the hull flows in the water and the hydrofoil is totally submerged in the water. As the speed of the boat increases and the AOA of the front and rear wing changes, the hydrofoils generate lift. At a certain speed, the lift generated by the hydrofoils equals the sum of the boat and pilot weights. As a result, the hull comes out of the water. Contrary to what happens in traditional boats due to pressure drag, the boat with hydrofoils lift the hull out of water without having an increase in drag with the increasing speed. The hydrofoils create a more efficient way of cruising. Decreasing the drag contributes to the better usage of the power needed for the movement of the boat. (Rosado, 1999)
The position of the boat can be adjusted by generating different lift of the front, rear and side wings separately. For instance, when the boat is unstable, the lower wing of two side wings experiences a bigger AOA and greater lift, while the higher wing experiences a smaller AOA and lower lift. As a result of these, the hydrofoil makes sure that the boat is kept stable. With the same working principle, the height can be ensured by adjusting the front and rear wings.

2.2.3 Control System

A control system is a system of devices, which commands, regulates, directs or manages the behavior of other device(s) or system(s) to achieve desired results. There are two major configurations of control systems, open loop control systems and closed loop control systems. (Shown in fig 2.2) In open loop control systems, output is only based on inputs. In closed loop control systems, not only current output is taken into consideration, but also corrections are made based on feedback. (electrical4u, 2011)



(Nise, 2011)

Figure 2. 2 Blocked diagrams of control system: (a) open loop. (b) close loop.

2.3.4 Sensors

In the previous student research, they decided to use the ultrasonic sensor to detect the height. However, there are two questions left. The first question is that the ultrasonic sensor selected by the previous researcher is not waterproof. The second question is that when the boat is sailing, the wave will definitely occur, which will affect the accuracy of the altitude (Shown in fig 2.3). The solution needs to be found out. For example, the accuracy can be improved by changing the type of the ultrasonic senor or changing the sample time in the software.

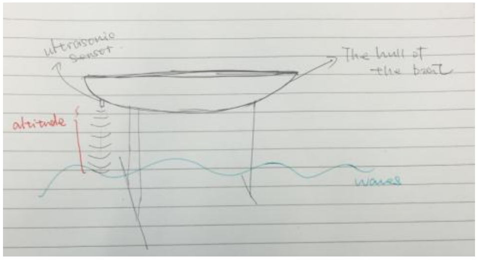


Figure 2. 3 The ultrasonic sensor measuring the altitude

2.3.5 Hardware and Software

In order to design a control system, Arduino (Shown in fig 2.4) is utilized as microcontroller.

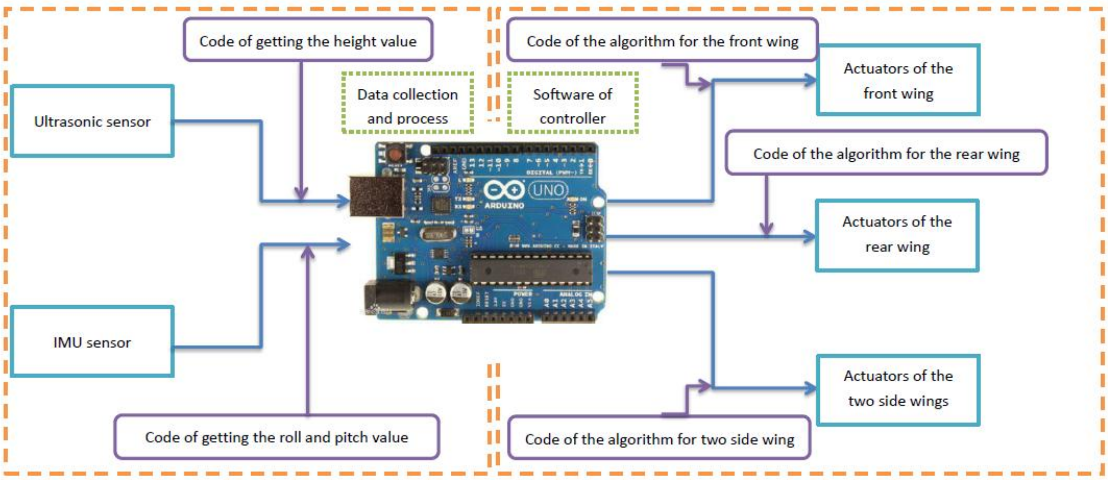


Figure 2. 4 Hardware and software

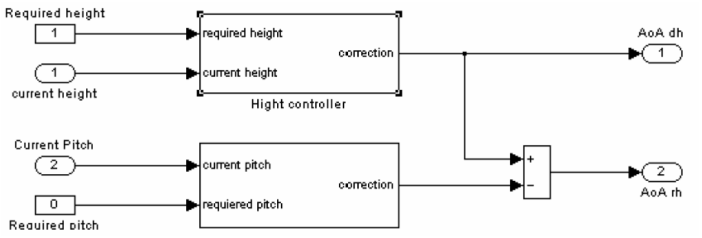
In order to design a control system, Arduino (Shown in fig 2.4) is utilized as microcontroller. All the projects built with Arduino follow a simple pattern called “Interactive Device”. This interactive device is an electric circuit. This circuit can contribute to detecting the environment via using sensors and converting the real-world measurements into electric signals. This allows the device to interact with the world by using actuators and electric elements. This pattern can convert an electric signal into a physical action. (Banzi, 2009)

As shown in figure 2.4, the hardware part in hydrofoil control system consists of four major parts: sensors, microcontroller, actuators and wings. The ultrasonic sensor and IMU sensor are used to

detect the position of the boat. The Arduino board works as microcontroller, which are used to collect and process the data according to the signals from the sensors. Besides, the control system will be designed and programmed with Arduino to control the actuators of the wings in hydrofoil.
For the software part (Shown in fig 2.4), the codes of achieving the height, pitch and roll value need to be implemented. Additionally, the codes of algorithm of PID controller for each actuator need to be designed and wrote to control wings separately. The controller will be realized by writing the code of algorithm with Arduino.

2.3.6 PID Controller

A PID controller continuously calculates an error value as the difference between a desired set point and a measured process variable. The controller attempts to minimize the error over time by adjustment of a control variable.
PID algorithm consists of three basic actions, the Proportional, the Integral and the Derivative actions. These 3 actions affect the plant in different ways; the proportional action responds immediately to changes in height error but is insensitive to how rapidly the error is changing. If the proportional gain is set high the steady state error will be small, but stability will be degraded and the response will tend to be oscillatory. The second action is derivative action, this adds an extra action that is proportional the rate of change of the error. It will improve high frequency and transient response but will highlight high frequency noise. The final action is integral action; this action removes steady state errors and improves low frequency response. However this then degrades stability and phase margins.
Therefore, the detailed PID algorithm needs to be designed and implemented in software as controller to realize the automatic control system (closed loop control system).



Structure of the twin P.I.D. controllers.

Figure 2. 5 Structure of the height controller (Miller, 2005)

This paragraph illustrates the general idea of PID control system (Shown in fig 2.5). The researcher will use this general idea to design the PID algorithm in particular. The details of PID algorithm are specified to determine that different tuning values (Kp, Ki, Kd) should effect on different data separately. In order to control the roll, pitch and height value at desired point, which data should be affected by which tuning value is the challenge to be met and is regarded as the problem.

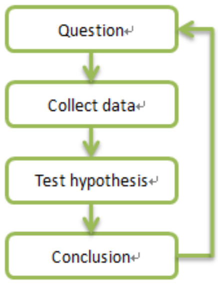
Chapter 3 Research Method

In this chapter, the research method and design method are described, including the explanation of why the certain methodology is chosen and research process.

3.1 Research Method

The first phase of the assignment is where the researcher will get familiar with the assignment, including a plan (Shown in Appendix P) about how to make this assignment a success. Additionally, the researcher starts self-studying the related knowledge which will be applied.

In this project, the researcher used the book called "Doing research – The Hows and Whys of Applied Research" as reference. The book provides a clear introduction to the research methods and statistics used in education, social work, and the social sciences. The emphasis is on applied research in which all the stages are demonstrated: setting up a research project, gathering data, analyzing the results, drawing conclusions, and preparing and evaluating research reports. (Verhoeven, 2011)



The method called "scientific method". (Shown in fig 3.1) It is a simple method which has four steps. First step is generating question; second step is collecting data; the third step is testing hypothesis and the last step is to come out a conclusion. According to the results, the researcher will raise new questions and start this flow again. The scientific method will be used in the tests with ultrasonic sensor, IMU sensor and test model to optimize the code of data collection, process and control system.

Figure 3. 1 Workflow of the scientific method

3.2 Detailed Research Method

In order to answer the research questions, questions are answered by the following way according to the table 3.1.

Research Question	Research Method
T1. What is the principle of making the boat run faster while keeping the boat stable? **T2.** Which method should be used to control the stability of the boat automatically?	Interview the project manager and clients Interview previous researcher Research on the hydrofoil technology Make a list of requirements Determine the functions and ideal situation
E1. What should be controlled by the pilot manually? **E2.** What else should be	Research on the existed hydrofoils Communicate with project manager and clients Present the current design

considered to keep the pilot safe under any conditions?	Improve from feedback Conclusion Analyze the result
E3. How to waterproof the sensors without affecting the signals?	Test the existed sensors Search another type Test
A1. How to reduce the error of the height measurement because of the waves? **A2.**How to optimize the accuracy of the data got from the sensors?	Research on the IMU sensor Research on the ultrasonic sensor Test the previous program Collect data Improve the codes of getting the height, pitch and roll value Test with the new codes Analyze the test results (Use scientific method)
A3. What is the range of the roll and pitch value of keeping the boat stable?	Research on the hydrofoil technology Design the controller Program the code Test with the test model to figure out the range of it. Analyze the result (Use scientific method) Improve the code with the test result
A4. How to check and adjust the AOA continuously to keep the solar boat stable when the hydrofoil is working? **A5.** How to check and adjust the AOA continuously to maintain the constant height when the hydrofoil is working?	Research on the control system Research on PID controller Design the PID algorithm (Specified to design different tuning values effect on different data) Program the PID algorithm with Arduino Build a test model Test with test model Analyze the result (Use scientific method)

Table 3. 1 Research method

3.3 Design Method

The design method chosen for this project is the v-model. (System/IT, 2011) The researcher is familiar with the delft design method and v-model. The reason why the delft design method was not selected is that a concept has been made already. The previous researcher has already selected the most suitable sensors and microcontroller by comparing many different kinds of sensors and microcontrollers, based on the delft design method. Now the new task is to use these sensors and microcontroller to design a control system and get it work. Therefore, the V-model is more suitable for programming projects than the Delft Design Method in this project.
The v-model (Shown in fig 3.2) gives a lot more structure for engineering a design into detail and getting it working. This is because of the explicit division into sub-systems and set moments for testing. V-model is a useful design method suitable for small to medium sized projects where

requirements are clearly defined and fixed. The researcher will follow the V-model to add the features which the client has requested at this semester.

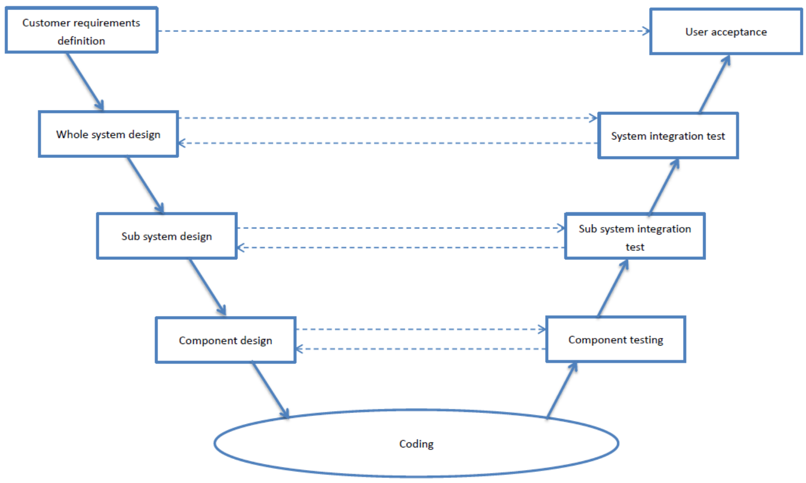


Figure 3. 2 V-model (Firesmith, 2013)

Based on the V-model, the workflow was listed in Appendix D.

3.3.1 Requirement Analysis

This is the first phase in the development cycle where the requirements are understood from the perspectives of clients. Underneath a list of all the tasks were worked on in this phase in no particular order.

Research proposal: This document includes all the results of the planning phase and serves as a starting point of the project;

- Get familiar to the C language and Arduino board.
- Understand the working principle of the hydrofoils.
- Understand the previous research and check the previous coding.
- Get familiar to the usage of ultrasonic sensor and IMU sensor.
- User requirements
- Literature research and archive studies
- Interviews with the previous researcher.
- Analysis of existing qualitative information
- Get familiar to how to use the ultrasonic sensor and IMU sensor

All of these contribute to achieve the success in the project. In the following phase, more details of process plan were presented as follow.

3.3.2 Whole System Design

In the system design part, the whole system of the project is defined. About the whole system in this project, the researcher needs to design the controller and program a whole algorithm to realize the controller. This is the first phase of the project where will actually start working on the problem. In this phase, the main idea and rough whole system will be designed. Underneath a list of all the tasks will be worked on in this phase:

- **System requirements:** The entire requirement for the systems will be defined in the list of requirements.
- **System overview:** A system overview will be made with all the interactions the system will have with the environment.
- **Test plan:** A test plan on system level will be written to see if the system meets the requirements.

3.3.3 Subsystem Design

In this phase, the whole control system of hydrofoil will be divided into smaller subsystems. A lot of work will be done to define the interfaces and requirements of each subsystems. Underneath a list of all the tasks will be worked on in this phase in no particular order:

- **System decomposition:** The hydrofoil control system will be divided into two systems. The advantage of this is that smaller parts can be engineered at the same time.
- **Subsystem requirements:** All the requirements for the systems will be defined in the list of requirements.
- **Test plan:** A test plan on each subsystem level will be written to see if the subsystem meets the requirements.

3.3.4 Component Design

The component design phase is similar to the subsystem design phase but one level deeper. At the start of this phase, the whole system will be divided into sub-systems and then design the component inside of each subsystem. Underneath a list of all the tasks will be worked on in this phase in no particular order:

- **Subsystem decomposition:** Each subsystem will be divided into several components. The advantage of this is that smaller parts can be engineered at the same time.
- **Component requirements:** All the requirements for the systems will be defined in the list of requirements.
- **Tests with available components:** Based on the scientific method, the accuracy of the sensors will be enhanced by testing and improving the software.

3.3.5 Coding

The actual coding of the system programmed in the design phase is taking up in the coding phase. The C language is chosen according to the previous program design. Based on the design phase, the algorithm will be programmed with Arduino.

3.3.6 Component Testing

In this phase, each component will be tested. These tests will be done to verify the components meet the requirements, which can contribute to determining and implementing in detailed design phase. If some requirements found out that are not met, the reason and improvement will be added on the existed design and coding.

3.3.7 Subsystem Testing

In this phase the subsystems will be tested according to the test plans for these sub-systems. These tests will be done to verify the system meets the requirements. During the process of subsystem testing with test model, some expected and unexpected results may come out. If some requirements found out that are not met, the reason and improvement will be added on the existed subsystem. Therefore, the testing contributes to optimizing the design.

3.3.8 System Testing

In this phase the whole control system will be designed with test plan to see if the system meets the requirements. Most of the software and hardware compatibility issues can be uncovered during system test execution.

3.3.9 User Acceptance

In order to accept by the user, a user manual will be designed. (Shown in Appendix B)

Chapter 4 Results

In this chapter, the results were described. Based on the process plan described in chapter 3, the structure of the results followed the structure of V-model. During the process of research and designing, the scientific method was utilized to optimize the result. (Shown in fig 3.1) The whole system design, subsystem design, component design, detailed design and testing were described in this chapter.

4.1 Concept-Design of the Whole System

The whole system design phase is the first step to design the project. Based on the functions and the requirements (Shown in Appendix C), an overview and main idea of the whole system were taken into consideration at first. In this phase, the system overview and test plan were described. The ideal situation and user manual were elaborated in Appendix B. The most important part is to describe all the inputs and outputs of the whole hydrofoil control system.

4.1.1 System Overview

First the coordinate was establishment. (Shown in Fig 4.1)

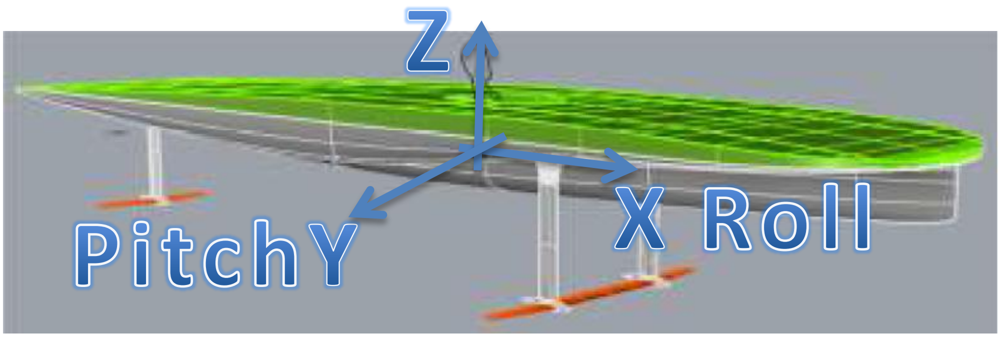


Figure 4. 1 Coordinate establishment

The boat with hydrofoils rotates around three axes. Rotation around the x-axes was called roll. Rotation around the y-axes was called pitch. The boat with hydrofoils must be stable and controllable around both 2 axes. The IMU sensor was used to achieve the data of roll and pitch value. The height value was achieved from the ultrasonic sensor. The stability of the boat was determined by the roll value. Through knowing the roll value, the system can calculate the angle of inclination of the boat. Besides, this error (angle of inclination) can be eliminated by the PID controller, which protects the boat from severe accidents when hydrofoil is working. The height of the boat was determined by the distance between the bottom of the hull and the surface of the water. Utilizing the height and pitch value, the PID controller can correct both the error of height and pitch to ensure that the whole hull stays perpendicular to the z-axis and maintain constant height. This whole system seems simple; however, the data collecting and accuracy, the PID algorithm and coding were three grand challenges.

Based on the ideal situation, requirements and the theory of control system, the whole system was designed as an automatic control system. The automatic control system was realized by the close loop control system. A closed loop system is also called a feedback loop control system. There are two common types of control system, one is open loop control system and the other one is feedback loop control system. In the feedback loop control system, current measurement of the output is taken into consideration and corrections are made based on the feedback. In the open loop control system, the output is only based on the input signal. The output of hydrofoil control system was not only based on the input signal, but also the difference between the current measurement and required set point. Therefore, the hydrofoil control system was feedback loop control system. The bellowing figure demonstrates the input, output and feedback of the whole hydrofoil control system. (Shown in figure 4.2)

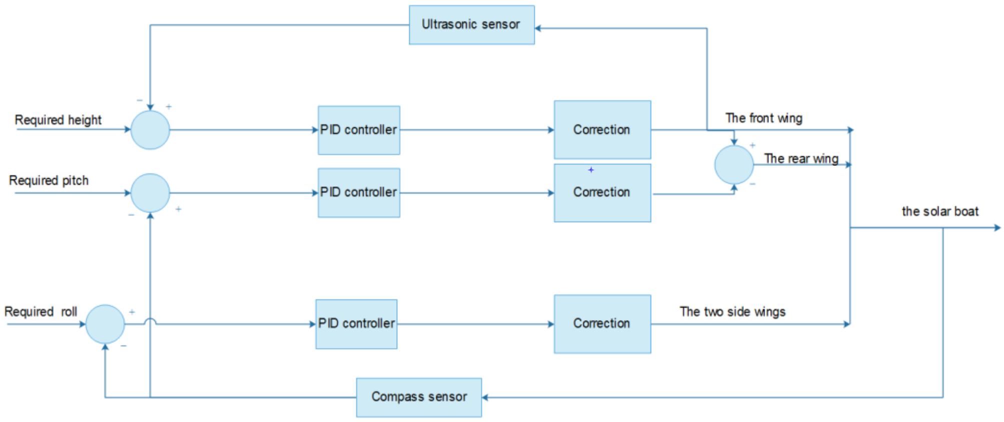


Figure 4. 2 Whole hydrofoil control system design

According to the previous, the ultrasonic sensor and IMU sensor was determined to be used. This hydrofoil of the solar boat haves four wings. Three wings are at front and one wing is at the back of the boat. The two sides wings at front for controlling the stability. The middle wings at front and the rear wing are for maintaining the height. The whole system was designed as automatic controlled system with self–regulation. Via using the feedback loop, eliminate the error in the system continuously. As mentioned before in the theoretical framework, the PID controller was used as the self-regulation. To begin with, the required value was set upped. Simultaneously, the sensor detected the current value. Following this, the feedback loop calculated the error. The PID controller corrected the error and sent pulse width modulation (PWM) value as output to control the actuators of the wings. At the same time, the sensor still detected the current value to measure the new error and started a new round of feedback loop. This system worked persistently. Measured the new error and corrected each error, which realized the dynamic stabilization in dynamic situation with self-regulation. Only when the hydrofoil knows the accurate position from the sensors, the whole system can be a stable and safe automatic control system. In view of this, the accuracy of the sensor was really essential. Furthermore, the PID controller was necessary for correcting the error. The detailed design of each PID controller, data processing and coding were presented in the following phase.

4.1.2 Test Plan of the Whole System

To check if the whole hydrofoil control system meets the requirements, the complete test plan can be found in Appendix I. This test plan was used to find out if the hydrofoil can work as the ideal situation.

4.2 Concept-Design of Subsystem

In the previous phase, the inputs and outputs of the whole system were defined. Following this, in the next phase, it is time to define sub-systems and interfaces between them. In the following paragraphs, subsystems design; subsystems overview, requirements and functions for the systems were presented, including the test plan.

4.2.1 Two Subsystems Design

The whole system was divided into two subsystems. They are stability subsystem and height subsystem. Combined these two subsystem, the hydrofoil can not only make the boat run faster, but also work at dynamic circumstance steadily and safely. Each subsystem has a specific set of functionalities. In the following paragraphs, each subsystem was described in details, specializing in requirements, functions and working theory.

4.2.2 Concept-Design of Stability Subsystem

The first subsystem was stability system. The function of the stability system was to ensure the pilot security when the hull is out of water. Through adjusting the two side wings continuously, the boat keeps stable at roll direction. Following this, what should be controlled? How to control it? These questions were answered in the practical life.

According to the intuitive experience, any persons can keep a stick up right on his finger. (Shown in Fig 4.3) There are two requirements why this can happen. One is that the eyes can observe the angle of inclination and the trend of inclination, the other one is that the finger can move to change the position of the stick. In practice, that is how dynamic stabilization works via using feedback loop.

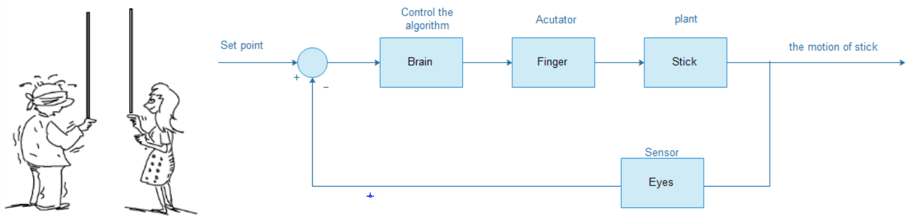


Figure 4. 3 Keep a stick up right manually

The dynamic stabilization of the hydrofoil used the same theory via using the feedback loop. The difference was that the sensors, PID algorithm and actuators replace eyes, brain and the finger.

Besides, the angular velocity was used instead of the trend of inclination. In conclusion, the IMU sensor was used to detect the angle of inclination and angular velocity to control the roll value. As a result of this, the stability automatic control system was designed. (Shown in fig 4.4)

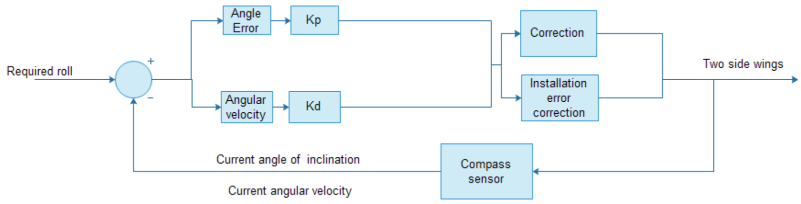


Figure 4. 4 Stability subsystem

The Kp in the diagram means proportional action. It corrected the new error. In this case, the new error was the difference between the set point of the required angle of inclination and the current angle of inclination. In the angle feedback control loop, the proportional action controlled the angle of inclination, which made the system react faster on deviations. The Kd in the diagram means differential action. It controlled the angular velocity, which had most of the time a stabilizing effect on the control loop. Accordingly, the parameter in the differential action was similar to the damping force, which restrained upon the oscillation of the boat.

The stability subsystem was based on the data got from the IMU sensor. The IMU sensor was detecting the angle of inclination so that the IMU sensor should be perpendicular to the z-axis of the boat strictly. In other words, if the IMU sensor does not install on the boat strictly, because of its character, the deviation of the angle will make the boat tilt to its side at roll direction with acceleration until severe accidents. The character of the IMU sensor was described in following paragraph 4.3 components design. In order to avoid this, installation error correction should be included in stability subsystem for robust system.

The output of the PID algorithm was PWM for adjusting the actuators. The PWM value kept steering and changing the AOA of the two side wings. When the boat is unstable, the lower wing experienced a bigger AOA and greater lift, while the higher wing experiences a smaller AOA and lower lift. As a result of these, the hydrofoil makes sure the boat stable. This is a correctional force that rolls the wings back toward level flight at roll direction. Based on figure 4.4, the electric circuit of stability subsystem was designed and connected. (Shown in Appendix E)

4.2.3 Test Plan of Stability Subsystem

For the subsystems test plans was made to check whether meet the requirements. This test plan can be found in Appendix G. The code of the stability subsystem was presented in K. A test model was built by the researcher to prove that this theory of control system works well. (See more details in paragraph 4.5 Test model.)

4.2.4 Concept-Design of Height Subsystem

Besides the stability subsystem, the other subsystem was height subsystem. The function of the height system is to ensure the hull out of water at constant height. As the result of this function, less resistance would be in the water so that the boat can sail faster. Through adjusting AOA of the front and rear wings, the boat maintains constant height and keeps stable at pitch direction.

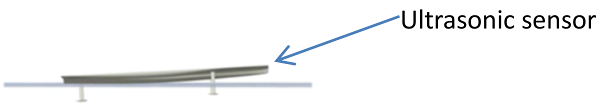


Figure 4. 5 Ideal situation B

As shown in the figure 4.5, the ultrasonic sensor would be installed at bow of the boat. The ultrasonic sensor detected the height value, which compared to the set point of the height value. Via adjusting the output PWM of the front and rear wings, the hull maintains constant height. However, only the ultrasonic sensor was not enough to keep the whole hull out of water. The ultrasonic sensor was installed at the bow of the boat; therefore, the height correction would cease until the first half part of the boat reached the set point of the height value. Just like the figure 4.5.

Figure 4. 6 Ideal situation C

In order to avoid the situation that only the first half part of the boat maintains the constant height, the pitch value should be contributed to the height subsystem. The ultrasonic sensor controls both the front and rear wing. The pitch value only controlled the rear wing to maintain the boat parallel to the water surface. (Shown in fig 4.6) Accordingly, the output PWM of the rear wing needs to be fusion process the error correction from the height and pitch.
Following this, the height subsystem was designed. (Shown in fig 4.7)

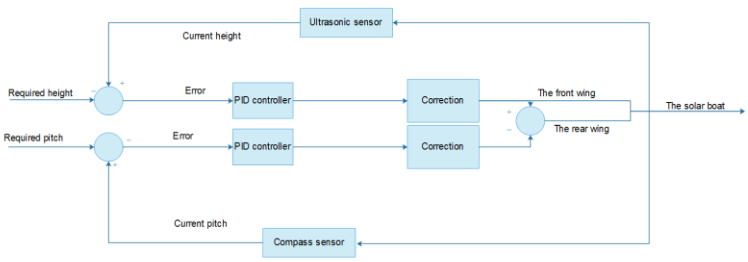


Figure 4. 7 Height subsystem

The height subsystem was based on the height and pitch got from the IMU sensor and ultrasonic sensor separately. Use the feedback loop of IMU sensor and ultrasonic sensor to correct the error of height and pitch separately. The height values controlled both the front and rear wings, while the pitch values only controlled the rear wing. The rear wing was controlled by the height and the pitch value. A little adjustment of the rear wing influenced the hull of the boat at pitch direction. This subsystem realized the function that the hull is out of water at constant height and also the whole hull is parallel to the water surface. The PID controller for correcting these two values was presented in the paragraph 4.3 component design. Based on figure 4.7, the electric circuit of height subsystem was designed and connected. (Shown in Appendix F)

4.2.5 Test Plan of Height Subsystem

For the subsystems test plans was made to check whether meet the requirements. This test plan can be found in Appendix H. The code of the height subsystem was presented in Appendix L.

4.3 Component Design

In this phase, the components of each subsystem were presented. Both in the stability and height subsystem, the components are measuring the position and the output PWM of actuators. Each of them was described in the following paragraph.

4.3.1 Components of Stability Subsystem

Based on the stability subsystem design, the method to control the stability of the hydrofoil was that control the PWM of the actuators, belonging to the side wings, through detecting the angular velocity and angle of inclination of boat at x-axis and the correction of the roll value. However, there are still two existed questions. How to measure these two kinds of value? How to control the output PWM?

The stability subsystem was divided into two components: one was measuring the angular velocity and angle of inclination; the other one was controlling the output PWM of two side wings including the installing error correction. Underneath the two components of stability subsystem was described. In this phase, the component was tested and proved that it can work well. That is to say, the component was improved and redesigned, based on the results of the test, to enhance the system. According to the chapter 4.2.2, the first important point of controlling the stability of hydrofoil is controlling the angle of inclination and angular velocity to correct the roll value.

4.3.1.1 Measure the Position of the Boat

In the whole system design, the closed loop control system and PID controller was selected. In this kind of control system, the current position was taken into consideration. In the stability subsystem, the roll value was controlled by the angle of inclination and angular velocity. Additionally, the velocity of the actuators was detected by hall sensor, which contributed to the control the position of the actuators and eliminated the installation error. In this phase, what was essential to be measured was presented in the following paragraph. The workflow of measuring the position was described in Appendix Q. The accuracy of IMU sensor was improved by programming and testing with the electric circuit (Shown in appendix E).

Roll

Only knowing the position continuously, the actuators can make accurate output. The IMU sensor Mpu6050 was utilized to detect both the roll and pitch value. There are accelerometer, magnetometer and gyroscope in the mpu6050 sensor. The roll and pitch value, the angle of inclination and angular velocity of the boat at x-axis can be known from this sensor. Additionally, mpu6050 can be easily communicated with Arduino. The Arduino board was used as microcontroller, which was determined from the previous research. The PID controller was programmed with Arduino, which contributed to control the actuators based on the error.

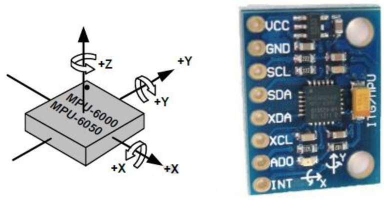


Figure 4. 8 Mpu 6050 IMU sensor

The mp6050 can detect the roll, pitch and yaw value. Rotation around z-axis called yaw.
The gyroscope can detect the angular velocity.
The accelerometer can detect the acceleration.
Both of these can detect in 3D axis. The figure 4.8 shows the compass sensor mpu6050.

Angular velocity

Via gyroscope in mpu 6050, the rotation angular velocity of x, y, and z-axis can be measured directly at the same time. This function was utilized to detect the angular velocity of boat in x-axis (roll direction). The following pictures show that the mpu6050 was measuring the angular velocity at x-axis.

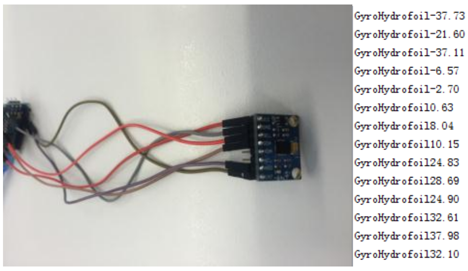


Figure 4. 9 Test picture of IMU sensor

The output of gyroscope is angular velocity, which was created by the motion of the boat. It has been tested. (Shown in fig 4.9) The noise of this signal is really small. These values were accurate enough. That is because that the motion of the boat does not influence the signal due to the gyroscope is detecting the motion. As a result of this, the angular velocity can be used directly without any filtering to know the trend of angle for keeping the hydrofoil stable at roll direction.

The detailed programming of how to get the angular velocity of x-axis was shown on the Appendix M.

Angle of inclination

The measurement of angle of inclination was not as easy as the angular velocity measurement. The first method was to measure the angle via using accelerometer.
Accelerometer can measure the acceleration caused by the gravity or motion of the object. The accelerometer in mpu6050 sensor can detect the acceleration of x, y, and z-axis. Based on these three values, each angle of inclination at different axis can be calculated. For instance, the angle of inclination at x-axis can be known by this formula

$$\text{angleX} = \text{atan}\left(\frac{\text{AccelermeterY}}{\text{AccelerometerZ}}\right) * 180/\pi$$

(Pedley, 2013)

The value of accelerometer Y and accelerometer Z can be easily achieved from the mpu6050 sensor. It seemed that angle of inclination at roll direction can be acquired. However the oscillation has great influence on the signal of acceleration, which makes it difficult to get exact angle of inclination. For the automatic control system, the accuracy of the sensor is the key to success. A test was done to see how much it would influence the angle of inclination.

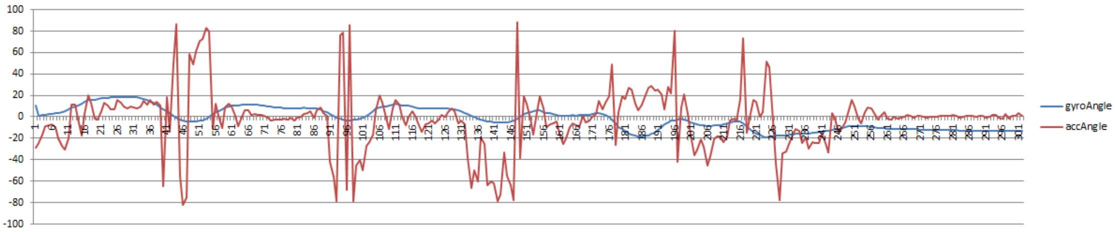


Figure 4. 10 Test of accelerometer and gyroscope

Above figure 4.10 shows the result of the test. The X-axis is sampling numbers and the Y-axis is the angle of inclination measured in degrees. The red line represents the angle calculated from the accelerometer. The blue line represents the angle calculated from the gyroscope.

This test consisted of two different test statuses. At the beginning, the tester oscillated the object manually. The front part of that chart demonstrates that oscillation part, which showed clearly that the sensitivity of the accelerometer caused great influence on the angle of inclination. Compared to the blue line, the noise of red line was severe when the object is oscillating. The oscillation caused interference signal, which overlay with the output signal. Because of that, the measured value from the accelerometer cannot reflect the angle of inclination exactly. The latter part of the test was under static situation. The tester fixed the IMU sensor at 0 degree. At the static situation, without any oscillating, the accelerometer worked well. As shown in the figure 4.10, the noise at that situation was not severe and the value was nearly zero, which was accurate.

Another method is to integrate the angular velocity got from gyroscope. According to this formula $d\theta = \omega * dt$ (Lauszus, 2012), the value of the angle of inclination can be calculated. The blue line represents the angle calculated from the gyroscope. However, the angle calculated from the angular velocity via integral operation can form integrating error. This error accumulates when the time is increasing. As shown in the front part of the chart, within a very short time frame, the gyroscope can reflect the angle of inclination well. The line is smooth. Nevertheless, the error accumulates constantly while the integral operation. As shown in the latter part of figure 4.10, the tester fixed the sensor at 0 degree, but the value calculated from the gyroscope in the figure was not 0 degree and it demonstrated a trend to decrease. It shows the fact that the angle of inclination calculated from the gyroscope accumulated when the time was increasing. The result from this test was that both the calculations from the accelerometer and gyroscope were not capable enough to detect the angle at real-time. According to the test, these two methods cannot reflect the angle of inclination at real-time.

In order to solve the accuracy of the calculation, Kalman filter was used to accurate the data. It is an algorithm, which uses a series of measurements observed over time, in this context an accelerometer and a gyroscope. These measurements will contain noise that will contribute to the error of the measurement. The Kalman filter will then try to estimate the state of the system, based on the current and previous states, that tend to be more precise that than the measurements alone. The code of Kalman filter is an open source online, which can found easily.

A similar test was done to see how the Kalman filter works. The X-axis is sampling numbers and the Y-axis is the angle measured in degrees. As demonstrated in the following figure 4.11, the green line represents the angle calculated from Kalman filter. The red line represents the angle calculated from the accelerometer. The blue line represents the angle calculated from the gyroscope. As mentioned before, gyroscope can reflect the accurate data within a very short time frame. The green curve overlaps on the blue curve perfectly. The noise in the green curve is much more less than in the red one. According to the chart, the Kalman filter applies smooth effects to the reflection of the angle. Besides, the accuracy is better than the two method mentioned before.

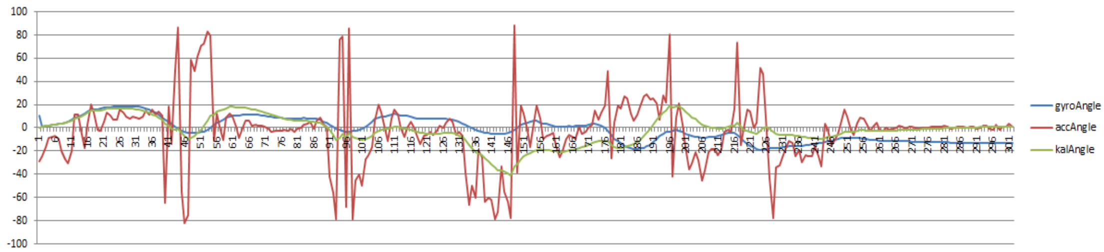


Figure 4. 11 Test of angle calculated from accelerometer, gyroscope and Kalman filter

In conclusion, the sensor mpu6050 was used. The gyroscope was used to detect the angular velocity of hydrofoil at x-axis. Both the measured value from accelerometer and gyroscope was used to calculate the angle of inclination of the hydrofoil at roll direction. During the process of calculation, the Kalman filter was utilized to enhance the accuracy of data. The angle of inclination and angular velocity was used as input signal for detecting the position of solar boat. These two kinds of input signal were capable of sending the correct position detected data. These data will be used in the PID algorithm to control the two side wings for controlling the stability of the boat. The code of angle of inclination was presented in Appendix N.

4.3.1.2 The Output PWM of Actuators

The accuracy of the two kinds of angle value was enhanced. According the PID algorithm, this system was designed. (Shown in figure 4.12)

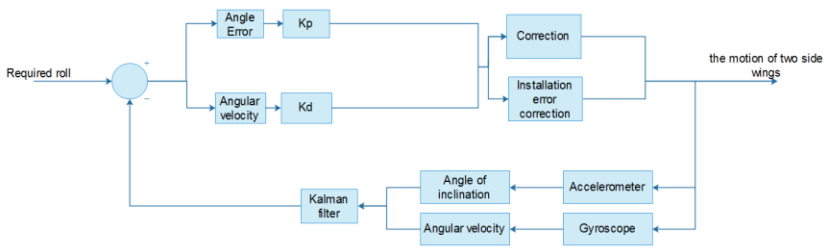


Figure 4. 12 Angle feedback loop

Based on the intuitive experience, the proportional action reacts to the instant error. In this feedback loop, the instant error was deviation between the required angle of inclination with the current one. The Differential action applies the suitable correction to the error previously, based on the changing trend of the error. In this case, the angular velocity represents the changing trend of error. The parameter of P and D action called tuning value. The only way of defining the tuning value was trying in practical life. How to find out these values was present in the test plan of stability subsystem. (Shown in appendix G)

In this phase, the accelerometer must be installed on the boat without any error. If there is any difference between the Z-axis of the accelerometer with the vertical direction of the boat, the value got from the accelerometer will have deviation. Because of the theory of how accelerometer sensor works. By measuring the acceleration caused by the gravity or motion of the object, the installation mistake will give rise to the accuracy problem. With the error of installation, the stability of the boat cannot be ensured. It would cause the boat go ahead at one direction in x-axis with acceleration until have a crash. So as to eliminate this problem, the installation error correction needs to be included in the whole stability system.

The installation error gave rise to the unstable situation. Only the Angle loop of hydrofoil stability control is not enough to ensure that the boat can keep stable. Therefore, the velocity loop of the hydrofoil stability control is necessary to add in the whole stability control system. The first step of controlling the velocity loop is to get the information from the actuator of the two side wings. For that, hall sensor was selected. Hall sensor can send a pulse when the motor rotates a single turn. The counter of the pulse and the time interval between the two pulses can calculate the speed and position of the two side wings. Acquiring the information of speed and position, PID algorithm were also designed to control the velocity of the two side wings automatically. The figure below demonstrates the whole stability system in details, including the detailed installation error correction.

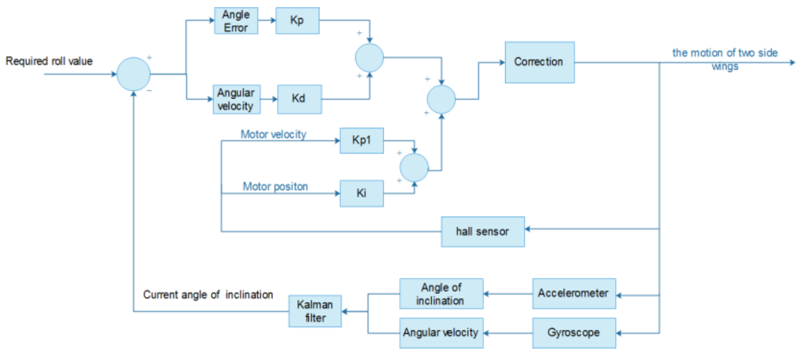


Figure 4. 13 Whole design of stability subsystem with detailed velocity controlled loop

The theory of velocity controlled loop was similar to the angle controlled loop. As shown in the figure 4.13, the proportional action Kp1 reacts on the velocity. The velocity was easily calculated from the hall sensor. The proportional action controls the velocity, which makes the system react faster on deviations. The integral action Ki controls the position of the motor, which eliminates the static deviation of the proportional action. The position of the motor was calculated by the accumulation of pulse of the hall sensor. There two tuning values need to be determined in practical life as well. The angle and velocity control loop superimpose with each other to control the output PWM of the two side wings. This closed loop via using feedback ensures that the hydrofoil can keep the boat stable. The angle closed-loop keep the boat tracing the balancing point. The installation error correction avoids the situation that the boat goes ahead at roll direction with acceleration, which improves the dynamic stabilization of the boat.

According to the whole designed figure (Shown in 4.14), the output PWM in the coding is:

PWM = (AngleBoat – Base)* AngleP + GyroBoat * Angle + Position* V_I + velocity * V_P

In this formula, Base is the set point. The value of Base was found manually from the mechanical balanced point. The detailed code was included in the stability subsystem coding in Appendix K. In conclusion, the whole detailed subsystem was designed with each component. The test plan of the stability control system through the test model was presented in Appendix J.

4.3.2 Component of Height Subsystem

Based on the height subsystem design, the method to control the height of the hydrofoil is that control the PWM of the actuators, belonging to the front and rear wings, through detecting height and the pitch value. However, there were still two existed questions. How to measure these two kinds of value? How to control the output PWM?
The height subsystem was divided to two components: one is measuring the height and pitch value; the other is controlling the output PWM of the front and rear wings. The accuracy of ultrasonic sensor was improved by programming and testing with the electric circuit (Shown in appendix F).
Underneath the two components of height subsystem was described.

4.3.2.1 Measure the position

In this phase, how to determine the right position of the boat was described. The height value controlled the AOA of the front and rear foils. By adjusting this, the height and pitch value of the boat would change. The ideal pitch value is zero, which means the hull was parallel to the water surface. The pitch value only controlled the AOA of the rear foil. The underneath paragraph describes the calculation of the height and pitch value in details. The workflow of measuring the position was described in Appendix III.

Height

The height value means the distance between the bottom of the hull and the water surface it is passing over. From the previous research in 2015, the ultrasonic sensor was selected as height detector. (Shown in the fig 4.14)

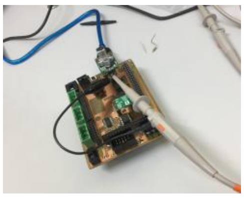


Figure 4. 14 Test of existed PCB board with old ultrasonic sensor

This ultrasonic sensor had been lying idle in the electrical lab for half year. A test with oscilloscope proved that it is broken. Above is the former ultrasonic sensor (shown in fig 4.14). The ultrasonic JSN-SR04T (Shown in fig 4.15) was selected as the height detector instead of the old one, because the waterproof part of the old one is tiny. Compare to ultrasonic JSN-SR04T, the old one cannot be easily installed on the boat with tiny waterproof part.

Figure 4. 15 Ultrasonic sensor JSN-SR04T

The ultrasonic sensor JSN-SR04T has a tiny chip which can easily interface with Arduino board. The distance detecting range is 15 ~ 200cm. The working temperature is -10 ~ 70℃. (Akhi, 2015) Those characters are suitable for the competition working environment. The transmitter of the ultrasonic sensor launches the signal and the receiver receives the signal. The ultrasonic sensor is controlled by a pulse. After the pulse sends, the sound wave launched from transducer. When the sound wave is emitted, the receiver waits until the transducer signal comes back to the ultrasonic sensor and gives a high signal at the output. Through the time difference and the propagation velocity of sound in the air, the travel distance and object distance can be calculated based on the following formula.

$$TravelDistance = (TravelTim \cdot SpeedOfSound)$$ (Lgel, 2013)

The above formula calculates the total distance that made the sound wave. This is thus the transducer, to the object and back to the transducer. The distance to the object, therefore, is the half of the travel distance.

$$ObjectDistance = \frac{TravelTime \cdot SpeedOfSound}{2}$$ (Lgel, 2013)

That is the theory about how the ultrasonic sensor works to detect the object distance. According to this theory, the height of the boat can be calculated.

In the chapter 2, the previous researcher selected ultrasonic sensor to detect the height. There are two questions left which need to be solved from the previous research. One is waterproof problem and the other one is accuracy. The first problem solved by selecting the JSN-SR04T sensor. This kind of sensor integrated with wire enclosed waterproof probe, suitable for wet situation. The other problem was accuracy. The wave occurs would affect the accuracy of the height. (Shown in Fig 2.3)

The problem was solved by the smoothing function in the software. The smoothing function is also called the circular buffer. This buffer is used to calculate a moving average. A moving average is useful to reduce the effect of waves. Underneath will show some details and explanations in this smoothing function. (Shown in fig 4.16)

```
const int num = 10;    // set up the sampling number

int readings[num];      // the readings from the analog input
int readIndex = 0;              // the index of the current reading
int total = 0;                  // the running total
int average = 0;                // the average

void loop() {

  total = total - readings[readIndex];// subtract the last reading:
  readings[readIndex] = analogRead(inputPin);// read from the sensor:
  total = total + readings[readIndex];// add the reading to the total:
  readIndex = readIndex + 1;// advance to the next position in the array:
  if (readIndex >= numReadings) {   // go to nect new round
    readIndex = 0;
  }
  average = total / numReadings;  // calculate the average:

}
```

Figure 4. 16 Smoothing function

The smoothing function can calculate the average of 10 values detected from the ultrasonic sensor. Compared with other average function, this smoothing function is calculating the average of 10 values while adding one new value and subtracting the first value each time. In other words, there is no lagging time in this smoothing loop. The ultrasonic sensor was detecting the new data continuously and calculating a running average. This smoothing function is useful for smoothing out the values from jumpy or erratic sensors. Even if the sensor gets abnormal value (in practical life, this abnormal value caused by waves), the average value is still approach to the last average value. Therefore, the abnormal value can be eliminated so as to enhance the accuracy. The code of the height measurement was presented in Appendix O.

The conclusion can be drawn that the moving average caused by the waves can be eliminated by the smoothing function and the ultrasonic sensor is waterproof. The two problems remained from the previous research has been solved.

Pitch

Besides the height value, pitch is another essential value which requires to be controlled. The height value was used to control the front and rear foils. The pitch value is used to control the rear foil which can keep the hull parallel to the water surface.

Similar to the roll value, pitch value can also be acquired from the mpu6050 sensor. As mentioned before, the accelerometer in the mpu6050 sensor can detect the acceleration of x, y, and z-axis. The vertical acceleration can be calculated from these three values. The noise and oscillation, effected on these values, can be eliminated via the Kalman filter. The whole component of detecting the pitch value was similar to the component of roll value. The only different aspect was that roll value keeps the boat stable around the x-axis, but the pitch value keeps the boat stable around the y-axis.

In conclusion, the sensor mpu6050 was used as well. The gyroscope in the sensor mpu6050 was used to detect the angular velocity of hydrofoil at pitch direction. The measured value from accelerometer and gyroscope was used to calculate the angle of inclination of the hydrofoil at pitch direction. During the process of calculation, the Kalman filter was utilized to enhance the accuracy of the data. The angle of inclination and angular velocity was used as input signal for controlling the position of solar boat, which was helpful to correct the pitch error. These data would be used in the PID algorithm to control the rear wing to make sure the hull of the boat parallel to the water surface when the bow met the height requirement.

4.3.2.2 The Output PWM of Actuators

The output PWM of actuators in the height subsystem was more complicated than the one in the stability subsystem. In the stability subsystem, the actuators of two side wings only controlled by the roll value. However, in the height subsystem, the front wings controlled by the height values while the rear wings controlled by the height and pitch values. First, the output PWM of the front wing was designed. (Shown in figure 4.17)

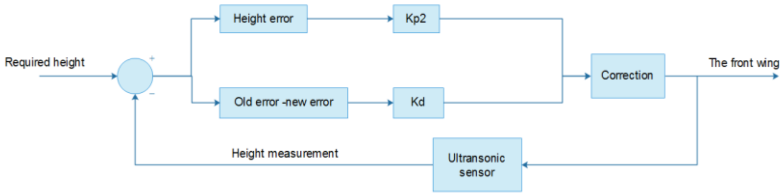


Figure 4. 17 The control system of front wing

As shown in the above figure, the actuator of the front wing was controlled by the height value. According to the formula of the PID controller:

Differential = (new error - old error) / time period
Proportional = new error (Miller, 2005)

The Kp2 in the figure means proportional action, it corrects the new error. In this height control subsystem, the error is the difference between the set point of the required height and the current height measurement. This error should be corrected by the proportional action. Additionally, the D in the figure means differential action. In the height feedback control loop, the proportional action controls the height error, which makes the system react faster on deviations. The differential action controls the height error impairment. As mentioned before in the stability subsystem, the differential action is similar to the damping force, which was helpful to restrain the vertical oscillation of the boat. The output PWM of the front actuator was designed in the following formula.

"PWMf = (Kp2)*Height_Error+(Kd2)*(New_Height_Error – Old _Height_Error)"

Old height error is the last loop of new height error. The output PWM of the PID algorithm was utilized to adjust the angle of attack to front wings. Via adjusting the angle of attack, the height of the boat can be controlled. However, only by adjusting the angle of attack of the front wing is not enough to make sure the whole hull of the boat maintain the same height with the water surface. The output PWM of the rear wing was design to help the hull parallel to the water surface.

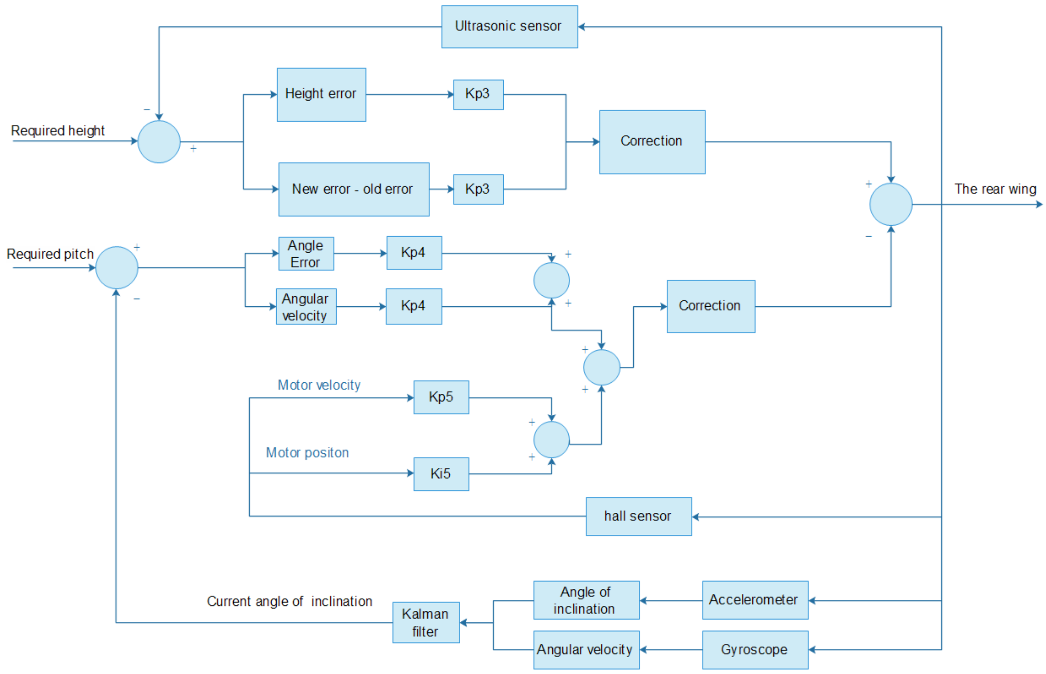


Figure 4. 18 The control system of rear wing

Above figure 4.18 demonstrates the design of the output PWM of the rear wing. All the ultrasonic sensor, hall sensor, accelerometer and gyroscope are utilized to detect the position of the boat. Expect the ultrasonic sensor; the other part of the rear wing system was the same as the stability subsystem. The theory of stability subsystem was controlling the roll value, while the theory of the height subsystem was controlling the pitch value. The Accelerometer and gyroscope was used to detect the angle of inclination and angular velocity at y-axis, which was contributed to control the pitch value. Furthermore, in order to eliminate the installation error of accelerometer, the velocity and position of actuators need to be controlled. Accordingly, the hall senor was used in the same way as the one in the stability subsystem.

On the contrary to the output PWM of the two side wings, the rear wing was not only controlled by the pitch value but also the height value. The pitch and height control loop superimpose with each other to control the output PWM of the rear wing. This closed loop via using feedback ensures that the hull of the boat was parallel to the water surface.
According to the whole designed figure (Shown in fig 4.18), The output PWM of the rear wing in the coding is:
"PWMb = (Kp3)*Height_Error+(kd3)*(New_Height_Error – Old _Height_Error)
- (Pitch_Boat – PitchBase)* (Kp4)
- GyroPitchBoat * (Kd4)
- RearPosition* (ki5)
-RearVelocity * (Kp5)"
In the formula, both the pitch and height value was used to control the output PWM of the rear wing. "Pitch Base" is the set point at 0 degree. The value of "Pitch_Base" was found manually from mechanical balanced point. All the tuning values need to be determined in practical life as well. How to find out these values was present in the test plan of height subsystem. (Shown in

appendix H) The detailed code was included in the height subsystem coding in Appendix L.

4.4 Coding

Based on the detailed design of PID algorithm, the code was implemented. (Shown in Appendix K, L, M, N, O) In the coding phase, some particulars were reported in this phase. These particulars contributed to meet the requirement of the whole design and connected each component.

4.4.1 Main Function of the Stability Coding

The code of stability subsystem was implemented by following the flow chart of the stability subsystem. (Shown in Appendix S) The flow chart of the stability subsystem was designed by combining each component design. In the programming, the sequence of functions was essential to make sure the design work. The order of functions was designed. (Shown in fig 4.19)

```
AngleBoatAvg[2] = AngleBoatAvg[1];
AngleBoatAvg[1] = AngleBoatAvg[0];
AngleBoatAvg[0] = AngleBoat;
AngleAvg= (AngleBoatAvg[0]+AngleBoatAvg[1]+AngleBoatAvg[2])/3;
if (AngleAvg <40 || AngleAvg>-40 ){
  MotorSpeedOutput();
  Deadtime();
  }
```

Figure 4. 19 Main function in the stability programming

After detecting the roll value of the boat and calculating the average of 3 values. The whole program went to judgment. When the average were smaller than 40 degrees, or bigger than -40 degrees, The “motorSpeedOutput()” function works. In this function, the error was calculated and the PID algorithm worked as a controller to send the output PWM value which can correct the error. Following this, it was “deadtime()” function. This one was helpful to limit the range of the calculation from the PID controller. Because of the range of the PWM in Arduino is 0~255. This main function of stability connected the two components of stability.

4.4.2 Main Function of the Height Coding

The code of height subsystem was implemented by following the flow chart of the height subsystem. (Shown in Appendix T) The flow chart of the height subsystem was designed by combining each component design. In the programming, the sequence of functions was essential to make sure the design work. The order of functions was designed. (Shown in fig 4.20)

```
AnglePitchAvg[2] = AnglePitchAvg[1];
AnglePitchAvg[1] = AnglePitchAvg[0];
AnglePitchAvg[0] = AngleHeight;
AngleAvg= (AnglePitchAvg[0]+AnglePitchAvg[1]+AnglePitchAvg[2])/3;
if (AngleAvg <40 || AngleAvg>-40 ){
  MotorSpeedOutput();
  Deadtime();
  }
```

Figure 4. 20 Main function in the height programming

In the height subsystem programming, the main function was similar to the one in the stability programming. (Shown in fig 4.20) After detecting the pitch value of the boat and calculating the average of 3 values, the whole program went to judgment. When the average were smaller than 40 degrees, or bigger than -40 degrees, The “motorSpeedOutput()” and “deadtime()” function works.

4.5 Test model

A test model was built by the researcher and a test-program was designed in order to prove that the feedback loop with PID controller is working according the design specifications. As shown in figure 4.22, the two-wheel car stands up-right on the ground by itself. This two-wheel car used the same theory of stability control system. It is an unstable system, which was similar to the hydrofoil system. The video of the working prototype was updated in YouTube. The link is https://www.youtube.com/watch?v=YColjHqtVpA

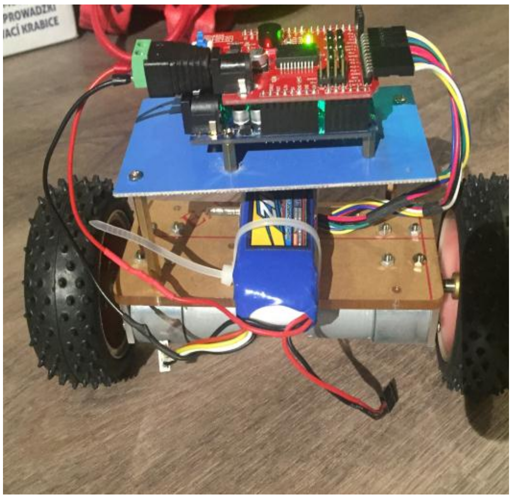


Figure 4. 21 Test model: two-wheel car

After several falling testing, the test model succeeds in standing up-right on the ground. The tuning values were found out in actual test by cut-and-trial. The test model was similar to the two side wings of the hydrofoil. Via controlling the roll values of the car, the feedback loop controlled the two wheels to keep the car stable on the ground. Additionally, the hall sensor was used to detect the velocity and position of the wheel to optimize the control.

The whole system of the car was the same with the hydrofoil stability control system. And the test model was proved that it can work well. Only with two wheels, the car can stay stable on the ground. Even a push on the car or working on rough ground, the car can still find it balanced point and stay up- right on the ground.

Chapter 5 Discussion

In this chapter, the research discussion is described below. These are based on the personal experiences achieved from this project and a critical looking back.

1. The previous researcher did not work at the company Epomat anymore as the researcher did. Sometimes, it was hard to understand the previous result and exchange ideas by e-mail. If working together, the research progress would be more efficient.
2. The blind range of ultrasonic sensor JSN-SR04T is 15 ~200cm. If the blind range reduce, it will be easier to control the hydrofoil.
3. The mechanical balanced point of the boat needs to be found in practice. This value should be changed in the coding. For instance, the researcher found the mechanical balanced point of the test model manually and set the value as set point in the coding. If the boat finished in building, the balanced point of the boat needs to be figured out.
4. The factor “time” played a too dominant role. A more realistic planning might have achieved a higher added value

Chapter 6 Conclusion and Recommendations

This section contains the general conclusion of this project and some recommendations about the further research, including the imperfection of the product.

6.1 Conclusion

In this phase, the main question and all the sub questions were presented in the following paragraphs. The whole detailed design was demonstrated in Appendix A.

Main research question: What is the best cost-effective design of the control system of a hydrofoil that can maximize the speed of the boat while keeping the boat stable and safe?
Answer: The feedback control system was used to control the hydrofoil. Via using the ultrasonic sensor and IMU sensor and PID controller, the height, roll and pitch value of the solar boat can be controlled. The detailed PID algorithm was designed and written with Arduino as a controller.

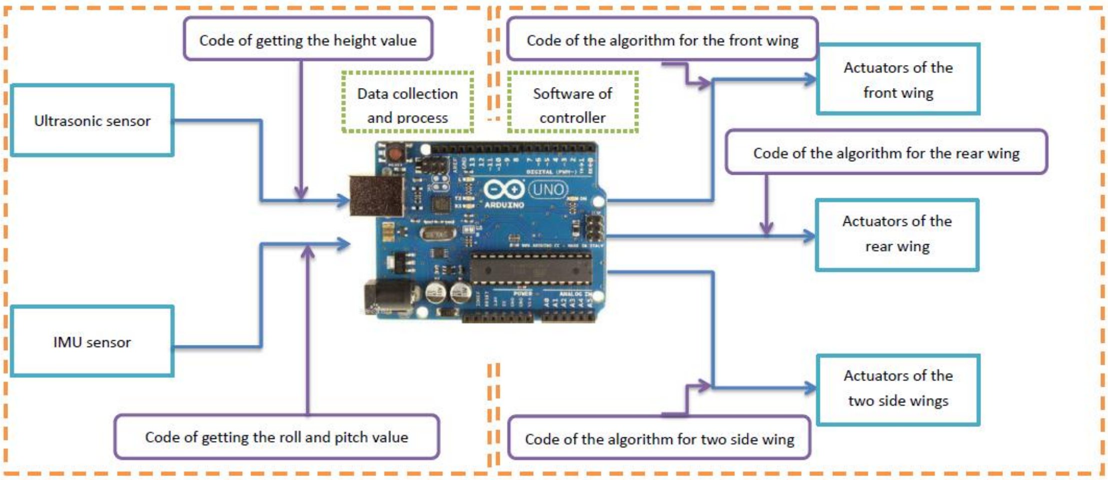


Figure 6. 1 Conclusion

As shown in figure 6.1, the controller was researched and the algorithm was designed. According to the algorithm of controller (Shown in Appendix A), the code was written. Besides, all the codes of gathering the data were written.
The accuracy problem of the sensors was solved. Smoothing function in the software enhanced the accuracy of ultrasonic sensor. The accuracy of the IMU sensor was improved by fusion the date processing from the accelerometer and gyroscope in IMU sensor.
Both the hardware and software was implemented. A test model was built to prove that the stability control system can work well. The height control system was designed with test plan.
With these two control system, the hull of the boat can be maintained at the constant height. As the result of hull out of water, the resistance was reduced so that the boat can run faster. For safety, the stability control system contributed to avoid severe accident.

Sub questions:

T1. How to make the boat run faster while keeping the boat stable?

Answer: Use the hydrofoil to lift the hull of the boat out of water, so that the resistance in the water was reduced and the speed increased. According to the calculation from the previous research, the maximum speed of the boat without hydrofoil is 17 km/h. The maximum speed of the boat with this hydrofoil is 24 km/h. Therefore, a control system needs to be designed to realize the hydrofoil technology.

T2. How to waterproof the sensors without affecting the signals?

Answer: This problem was solved. In the previous research, the old ultrasonic sensor only has short waterproof part. (Shown in fig 4.14) Compare to the new ultrasonic sensor, it was not easily installed on the boat. (Shown in fig 4.15) Additionally, the new ultrasonic sensor integrated with wire enclosed waterproof probe, more suitable for wet situation.

T3. Which method should be used to control the stability of the boat automatically?

Answer: Closed loop control system. The detailed design was shown in figure 4.7. The closed loop works with PID controller can realize the stability of the boat automatically.

E1. What should be controlled by the pilot manually?

Answer: There should be a button or display screen to turn on and turn off the control system of hydrofoil.

E2. What else should be considered to keep the pilot safe under any condition?

Answer: Once the roll value of the boat is bigger than 40 degree or smaller than -40 degree (This value was known by testing with test model), the actuators would be shut down automatically. Additionally, a so called "deadman-switch" would be used.

A1. How to reduce the error of the height measurement because of the waves?

Answer: This question was actually another deeper question about the accuracy of the data got from the ultrasonic sensor. In practice, the wave influenced the detecting accuracy. Therefore, the researcher reduced the error by adding the smoothing function in programming.

A2. How to optimize the accuracy of the data got from the sensors?

Answer: In this whole system, two kinds of sensor were used to detect the boat position and collect the data. The data are height, pitch and roll value. Two aspects were taken into consideration to enhance the accuracy of the data got from the sensor. They were hardware and programming in software. At the end of the project, the accuracy of ultrasonic sensor and IMU sensor were improved and can be proved to use to detect the boat position. The accuracy was improved by coding, testing, analyzing the test results and optimizing the program.

A3. What is the range of the roll and pitch value of keeping the boat stable?

Answer: In the programing, the range of the roll and pitch value is -40 degree to 40 degree. It was proved in the test model that can be used.

A4. How to check and adjust the AOA continuously to keep the solar boat stable when the hydrofoil is working?

Answer: In the control system, via setting the mechanical balance point of the boat, and compare it with the current position detected from the IMU sensor. The difference was the error in the closed loop. The PID controller contributed to eliminate this error. That was the way to keep the boat stable. (Shown in fig 4.4) The detailed PID algorithm, especially in tuning values (Shown in Appendix A), was designed and written as the software with Arduino. This works as a controller.

A5. How to check and adjust the angle of attack continuously to maintain the constant height when the hydrofoil is working？

Answer: Closed loop control system. Use the similar control theory of the stability control system. The height value was utilized to control the front and rear wings, and the pitch value was utilized to control the rear wings, so that the boat can maintain at the constant height and the hull was parallel to the water surface. The detailed PID algorithm, especially in tuning values (Shown in Appendix A), was designed and written as the software with Arduino. This works as a controller.

6.2 Recommendations

Due to lack of time and the hydrofoil was not finished in building, only the stability control system was tested. The height control system has not been tested. Therefore, the recommendations can contribute to further design.

Height controller

This part of height controller was designed based on the many existed hydrofoil control system. It will be better to build a test model to prove that the design of PID controller in the height subsystem can work in practice.

Tuning values

The stability subsystem was tested in the test model and proved that it can work now. But the tuning values of the hydrofoil need to be found out in sea trials in practice. The tuning values of the height controller also need to be found out in practice via sea trial.

Actuators

When the hydrofoil is finished in building, a suitable motor should be selected with enough torque. Therefore, the current of the motor will also increase. As a result, a new drive chip needs to be used for realizing the electric circuit.

Reference List

Akhi. (2015, December 15). Retrieved April 8, 2016, from tokopedia: https://www.tokopedia.com/akhishop/jsn-sr04t-waterproof-ultrasonic-sensor

Banzi, M. (2009). *Geting started with Arduino.* Dale Dougherty.

Dottoresso. (2008, April 25). *Nature-Technology*. Retrieved 4 19, 2016, from Flickr: https://www.flickr.com/photos/16997472@N03/2454894041/

electrical4u. (2011). *Conntrol system*. Retrieved 2 25, 2016, from electrical4u.com: http://www.electrical4u.com/control-system-closed-loop-open-loop-control-system/

Epomat. (2015). Retrieved February 15, 2016, from Epomat: http://www.epomat.com/site/engineering

Firesmith, D. (2013, November 11). *SEI insights*. Retrieved February 17, 2016, from Using V-model for testing: https://insights.sei.cmu.edu/sei_blog/2013/11/using-v-models-for-testing.html

Kagendijk, L. J. (2015). *Hydrofoil design.* leeuwarden.

Lauszus, K. S. (2012). *A practical approach to Kalman filter and how to implent it*. Retrieved 2 10, 2016, from TKJ electrics: http://blog.tkjelectrics.dk/2012/09/a-practical-approach-to-Kalman-filter-and-how-to-implement-it/

Lgel, I. (2013). Retrieved April 10, 2016, from Teachengineering: https://www.teachengineering.org/activities/view/nyu_soundwaves_activity1

Miller, C. (2005). *Control system for international Moth on hydrofoils.* Glasgow University.

Nise, N. (2011). *Control systems engineering.* Wiley.

Pedley, M. (2013, March 6). *Tilt sensing using a three axis accelerometer.* Retrieved April 11, 2016, from http://www.nxp.com/files/sensors/doc/app_note/AN3461.pdf

Radio corpotation of America. (1963). *A mechanical hydrofoil control system preliminary design study program.* Washington: Bureau of ships.

Rosado, T. (1999). Retrieved 2 11, 2016, from http://web.mit.edu/2.972/www/reports/hydrofoil/hydrofoil.html

System/IT. (2011, 11 4). Retrieved 2 22, 2016, from http://crackmba.com/v-shaped-model/

Veen, E. v. (2015). *Regelsysteem voor Draagvleugels.* Leeuwarden.

Vellinga, R. (2009). *Hydrofoils .* United States: Peacock Hill.

Verhoeven, N. (2011). *Doing research. The hows and whys of applied research.* Amsterdam: Boom Lemma.

Appendix

Appendix A: Whole system design

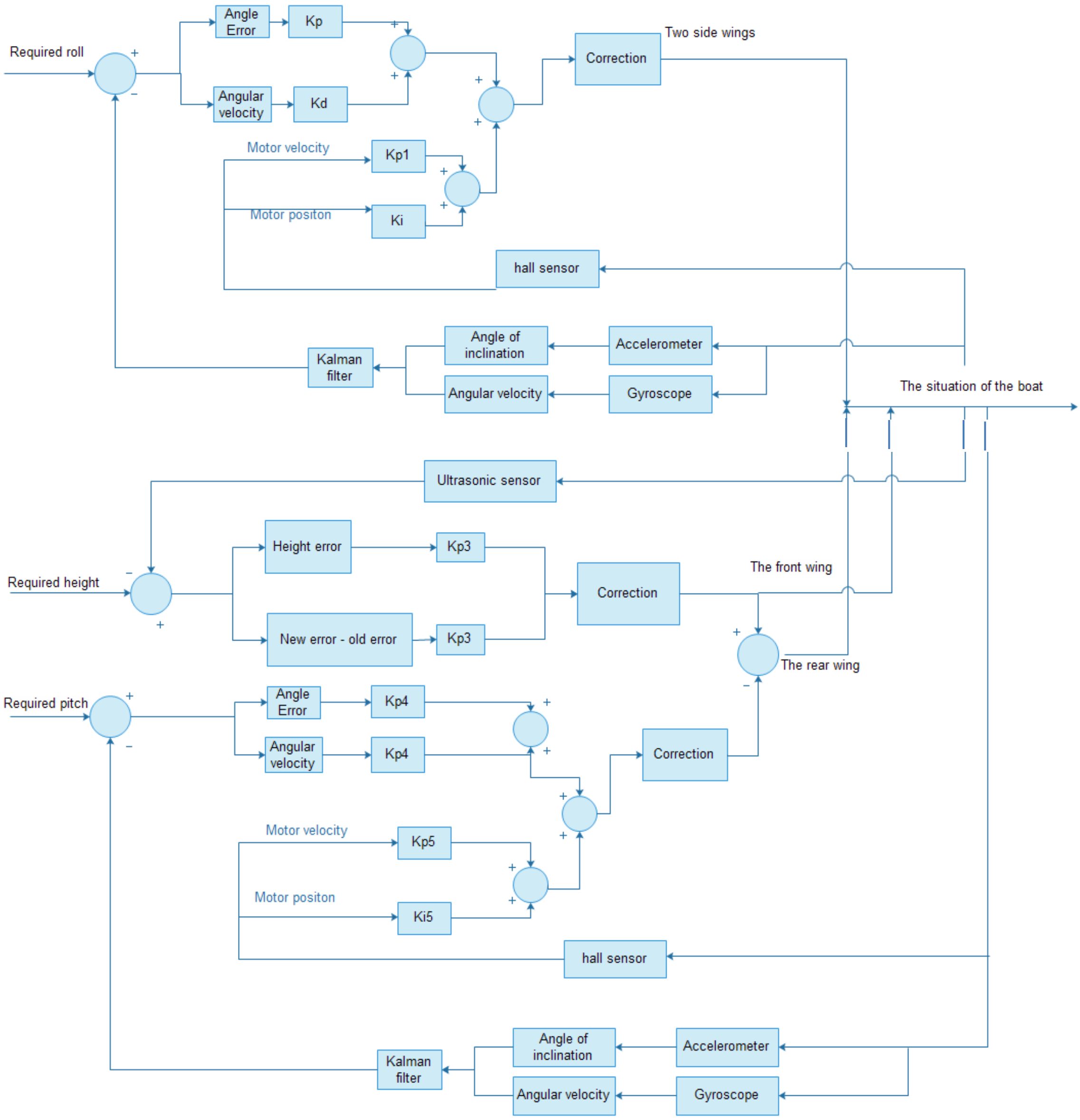


Figure 1 Concept-Design of whole control system

The whole control system was designed (Shown in figure 1). Based on the general idea of whole system (Shown in figure 2), the researcher was specified to design the tuning values (Kp, Ki, Kd). Different tuning values should be effect on different data. According to this concept-design of the whole control system, the algorithm can be written.

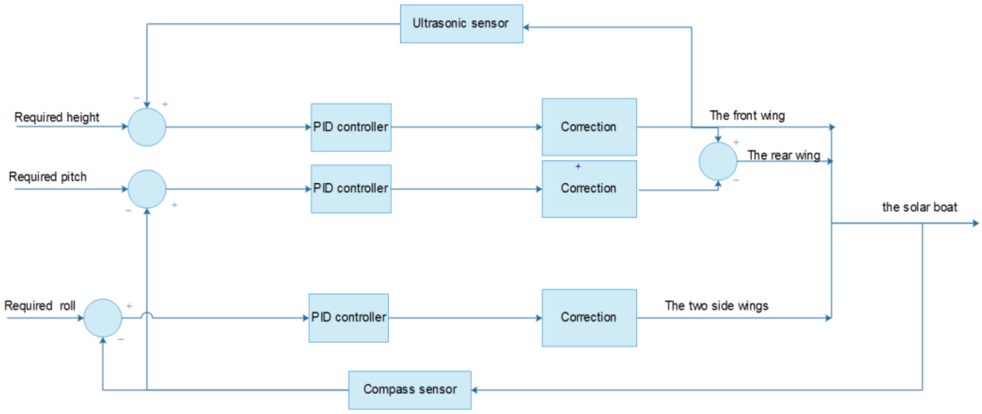


Figure 2 General idea of whole system

Appendix B: User manual

Figure 3 Ideal situaion

A B C

1. The ideal situation of the hydrofoil control system is that when the boat is sailing (Shown in Fig 3A), the pilot takes off the hydrofoil manually. (Shown in Fig 3B)
2. When the height value is more than 20cm, the pilot pushes the button to turn on the Height controller and stability controller.
3. After taking off, the boat keeps stable and maintains at constant height. (Shown in Fig 3C) The idea situation of stability is that both the roll value is at 0 degree. By lifting the boat out of water, the friction of the boat decreased. The hydrofoil control system ensured the dynamic stabilization of the boat, constant height to speed up and the safety when hydrofoil is working.
4. If the pilot wants to land the boat on the water, turn off the height controller first and adjust the front and rear wing to lower AOA manually.
5. As result of the lower AOA, the height will reduce. When the height is smaller than 20 cm, the pilot pushes the button to turn off the stability controller.
6. The boat lands on the water and continues to flow and sail on the water.

Appendix C: Requirements and functions

Requirements

The goal of hydrofoil control system is to make the boat run faster. In order to realize this, the hydrofoil needs to lift the hull of the boat out of water and maintain the stable and constant height. In order to reach this goal, a list of requirements needs to be met.

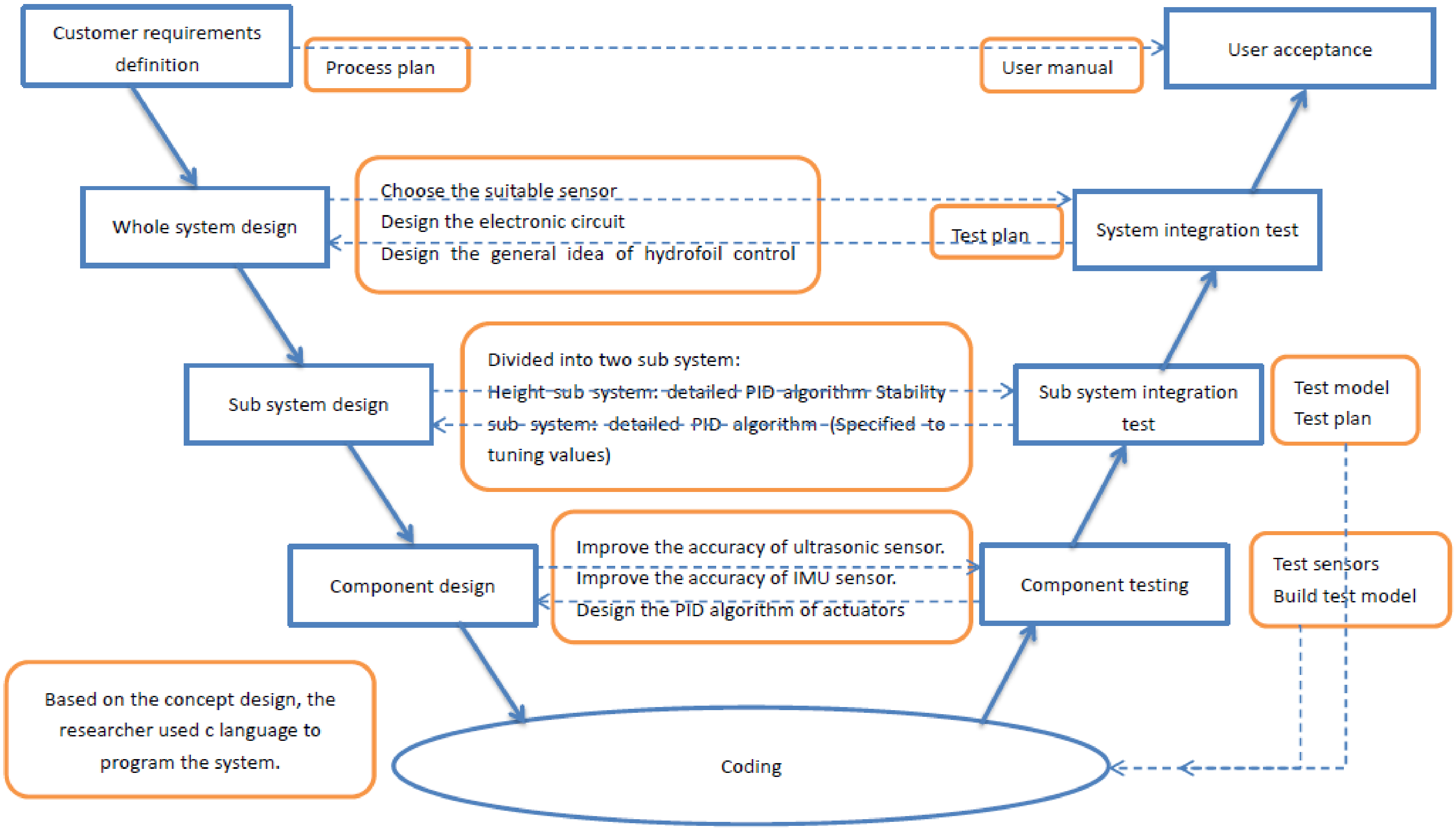
1. Steady and safe.
2. Control the two side wings of hydrofoil for stability.
3. Control the front and rear wings of hydrofoil for height.
4. The accuracy of the IMU sensor.
5. The accuracy of the ultrasonic sensor.
6. The waterproof system.
7. Low power consumption.
8. User friendly.

Functions

The functions of the whole control system are:

- The hydrofoil can keep the boat stable when taking off.
- The hydrofoil can keep the boat stable when sailing.
- The hydrofoil can find the balanced point of the boat again after turning around.
- The hydrofoil can reach the bow at desired height when taking off.
- The hydrofoil can keep the hull of boat at constant height, which also realizes level sailing.
- The hydrofoil can maintain the hull of the boat at constant height at unstable situation.
- The control system will automatically shut down when the boat falls on the water.

Appendix D: Workflow



Appendix E: Electric Circuit of Stability Subsystem

The electric circuit of stability subsystem was designed and connected. (Shown in fig 4) The layout of electric circuit was drawn. (Shown in fig 5)

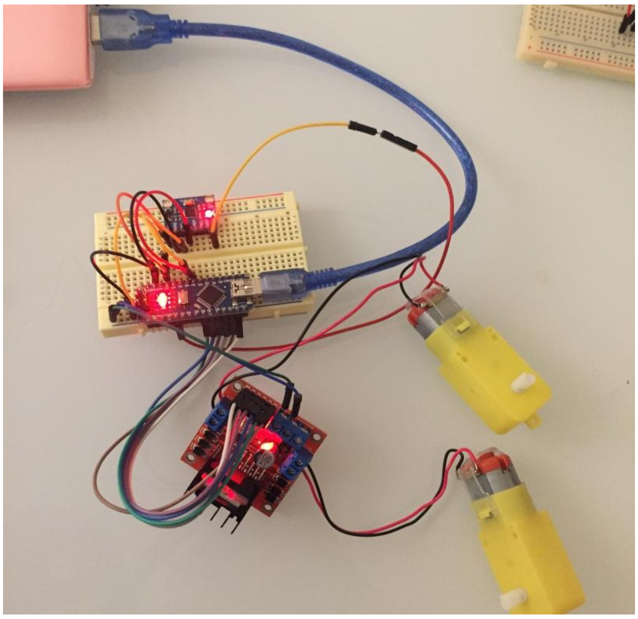


Figure 4 Electric circuit of stability subsystem

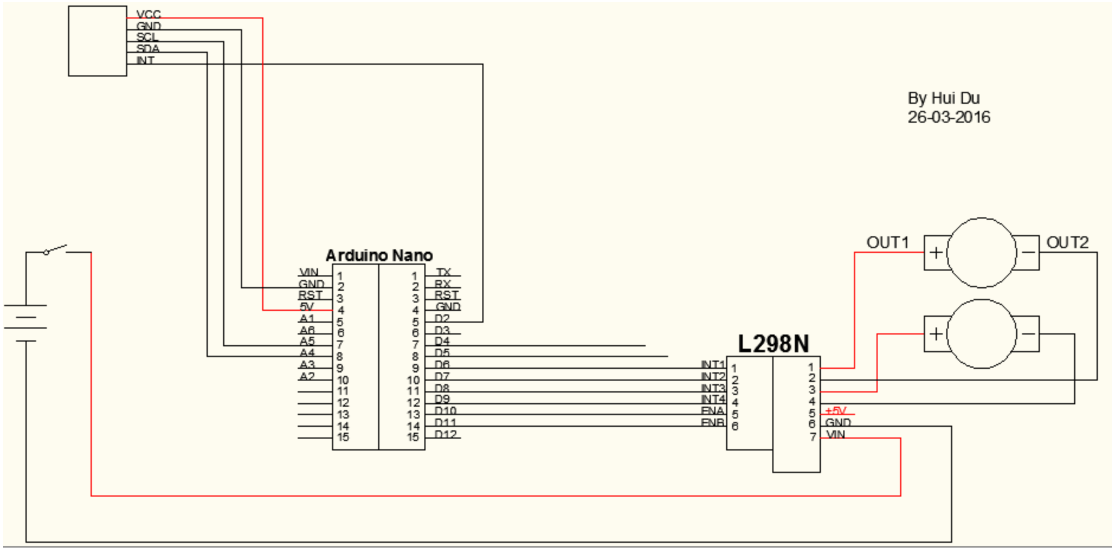


Figure 5 Electric circuit layout of stability subsystem

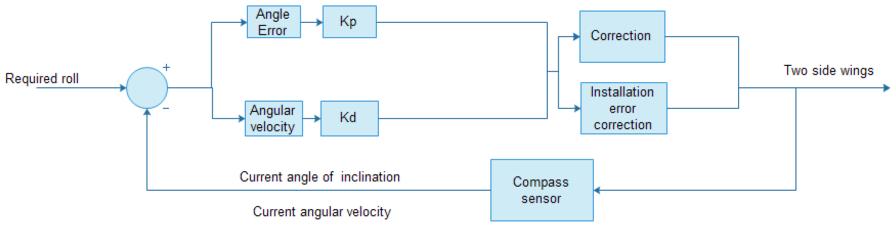


Figure 6 Concept-design of stability subsystem

Based on the concept-design of stability subsystem (Shwon in fig 6), IMU sensor, Arduino board, drive chip (L298N) and two motors are connected. The detailed PID algorithm (Shown in Appendix A) was implemented with the Arduino by writing the program.

Appendix F: Electric Circuit of Height Subsystem

The electric circuit of height subsystem was designed and connected. (Shown in fig 7) The layout of electric circuit was drawn. (Shown in fig 8)

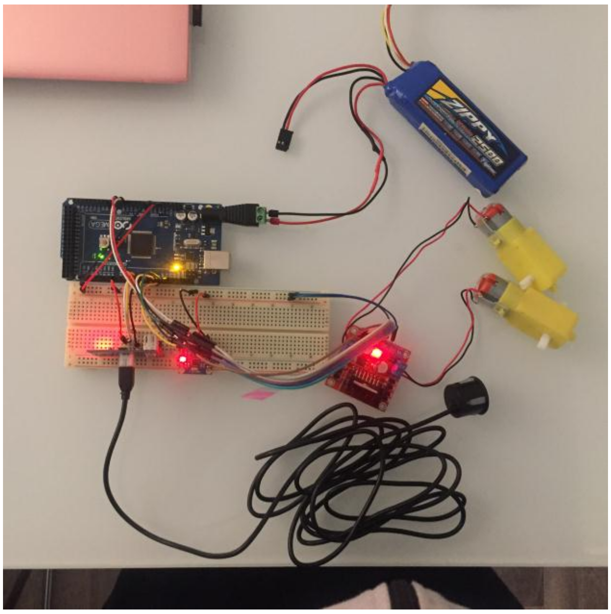


Figure 7 Electric circuit of height subsystem

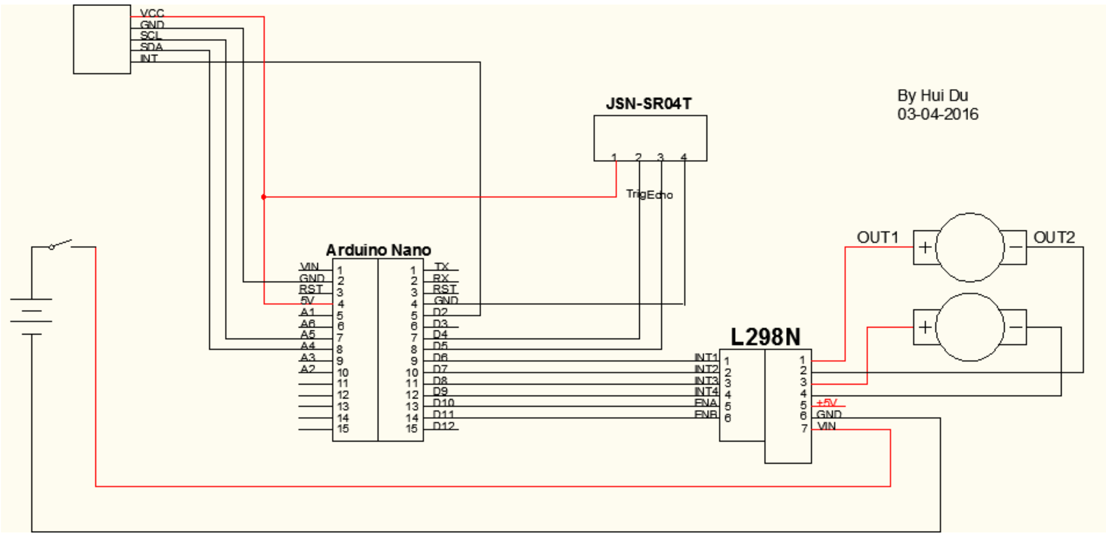


Figure 8 Electric circuit layout of height system

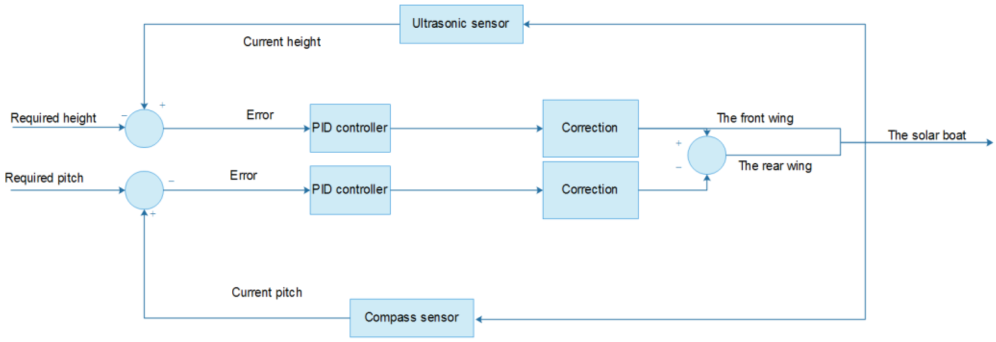


Figure 9 Concept-design of height subsystem

Based on the concept-design of height subsystem (Shown in fig 9), IMU sensor, waterproof ultrasonic sensor, Arduino board, drive chip (L298N) and two motors are connected. The detailed PID algorithm (Shown in Appendix A) was implemented with the Arduino by writing the program.

Appendix G: Test plan of stability subsystem

Test the stability subsystem of hydrofoil when it is sailing.

1. Introduction

This test plan was written as a reference to test the stability subsystem in the hydrofoil. This subsystem only needs to be tested when the boat is sailing out of water.

2. Aim

The aim of this test plan is to prove that the code of the stability subsystem can work well. This code need to be proved that it can realize dynamic stabilization. The requirements were listed as follow.

- The hydrofoil can keep the boat stable at roll direction.
- The hydrofoil can keep the boat stable at unstable situation.
- Dynamic stabilization.
- The control system will automatically shut down when the boat falls on the water.

3. Functionality to be tested

- The hydrofoil can keep the boat stable by itself when the hull is out of water.
- The hydrofoil can keep the boat stable by itself at unstable situation.
- After the boat turning around, the boat can still find its balanced point and maintain stable.
- The control system will automatically shut down when the boat falls on the water.

4. Hypotheses

Assuming that all the functions of this subsystem work well, then the hypothesis should be: The hydrofoil can keep the boat stable at roll direction when the boat is taking off and sailing. When the boat is turning around, the boat will tile to one side. After the boat turning around, the boat can still find its balanced point and maintain stable. Additionally, when the hull falls on the water, the actuators of hydrofoil will not move any more.

5. Target group

The tester should have basic knowledge of PID controller and the Arduino programming language. The tester should also be familiar with fluid dynamics and the characters of all the components in this solar boat, like motors, solar panel and so on. The rescuer should be standby to make sure the safety of the pilot.

6. Actual actions during the test

1) Before testing by sea trial, the boat with hydrofoil should be calculated the following things:
    i. Take off velocity and maximum velocity when hydrofoil is working, which contributes to limit the speed when sailing.
    ii. The desired height value and set it as set point in the programming.
    iii. The center of gravity, the center of buoyancy and mechanical balanced point.
2) Before testing by sea trial, the boat with hydrofoil should be made sure this following things:

i. The center of gravity with pilot is behind the center of buoyancy. When the hydrofoil is not working, the stem should be buoy higher than the stern. This permits a takeoff with the stem up and a tiny additional positive AOA on the wings.

ii. The IMU sensor should be installed on the boat, and its front side is perpendicular to the z-axis of the boat.

3) Set the mechanical balanced point as roll value in the programing.
4) Prepare the battery of 12V and 5V and test it on the day with sunniness.
5) Charge the actuators and Arduino board separately.
6) Take off the boat manually. Keep the power low to limit speed.
7) Turn on the power of Arduino.
8) If the hydrofoil cannot keep the boat stable when taking off, change the value of proportional action on the angle loop in the programming.
9) Disconnect all the batteries and connect the Arduino board with computer.
10) Upload the new program to the Arduino board and then repeat step 5-10 until the hydrofoil can keep the boat stable when taking off.
11) If the hydrofoil keeps the boat stable at roll direction with oscillation, change the value of differential action on the angle loop in the programming.
12) Disconnect all the batteries and connect the Arduino board with computer.
13) Upload the new program to the Arduino board and then repeat step 5-13 until the hydrofoil can keep the boat stable without oscillation when taking off.
14) After taking off, if the hydrofoil cannot keep the boat stable when sailing, adjust the value of proportional action on the angle loop in the programming a little.
15) Disconnect all the batteries and connect the Arduino board with computer.
16) Upload the new program to the Arduino board and then repeat step 5-16 until the hydrofoil can keep the boat stable when sailing.
17) After taking off, if the hydrofoil keep the boat stable with oscillation when sailing, adjust the value of differential action on the angle loop in the programming a little.
18) Upload the new program to the Arduino board and then repeat step 5-18 until the hydrofoil can keep the boat stable when sailing.
19) Turn around the boat, the boat will banking.
20) If the boat cannot find its balanced point after turning around, change the value of proportional and integral action on the velocity loop in the programming by trial and error to optimize the dynamic stabilization at unstable situation.
21) Disconnect all the batteries and connect the Arduino board with computer.
22) Upload the new program to the Arduino board and then repeat step 5-22 until the hydrofoil can find the balanced point of the boat after turning around.
23) Land the boat on the water.
24) Take off the boat manually and sail it at maximum speed.
25) Repeat step 7- 22. Change the tuning values in programming by trial and error to optimize the dynamic stabilization.
26) Land the boat on the water.
27) Charge the actuators and Arduino board with solar supply.
28) Repeat step 24-26.
29) Get the boat with hydrofoil out of water and record the turning values.

7. Average and bandwidth

Best value: The hydrofoil can keep the boat stable when the hydrofoil is taking off and sailing. Additionally, the hydrofoil can find the balanced point of the boat after turning around. The control system of hydrofoil should shut down automatically when the boat falls on the water

Bad value: The boat cannot keep stable all the time.

Worst value: The boat falls on the water and breaks while testing.

8. Predicted conclusions

Desired case

The desired case is that the hydrofoil can keep the boat stable when the hydrofoil is taking off and sailing. Additionally, the hydrofoil can find the balanced point of the boat after turning around.

Other cases

The possible problems in the process of implementing the stability control system were listed as follow. Besides, the corresponding possible reasons which will be helpful to solve this kind of problems were included.

Case 1: No power supply

- Change the battery.

Case 2: The hydrofoil cannot keep the boat stable on the water

- The tuning value of proportional action needs to be changed.

Case 3: The hydrofoil can only keep the boat stable on the ground only for a short time.

- The tuning value of differential action on angle loop needs to be changed.
- The tuning values of proportional and integral action on velocity loop need to be changed.

Case 4: The boat cannot keep stable all the time

- The power supply from solar panel is not steady.

Appendix H: Test plan of height subsystem

Test the height subsystem of hydrofoil when it is sailing.

1. Introduction

This test plan was written as a reference to test the height subsystem in the hydrofoil. This subsystem only needs to be tested when the boat is sailing out of water.

2. Aim

The aim of this test plan is to prove that the code of the height subsystem can work well. This code need to be proved that it can realize the function of height subsystem. The requirements were listed as follow.

- The height controller can keep the bow at constant height.
- The pitch controller can keep the stern at constant height. In other words, the hull is parallel to the water
- Dynamic stabilization at pitch direction.
- The control system will automatically shut down when the boat falls on the water.

3. Functionality to be tested

- The hydrofoil can maintain the hull of the boat at constant height at unstable situation.
- The control system will automatically shut down when the boat falls on the water.

4. Hypotheses

Assuming that all the functions of this subsystem work well, then the hypothesis should be: The hydrofoil can keep the bow at constant height when the boat is sailing. When the boat is taking off, the hydrofoil let the bow of the boat reach the desired height first, and then, rise the stern up to realize level sailing. Additionally, when the hull falls on the water, the actuators of hydrofoil will not move any more.

5. Target group

The tester should have basic knowledge of PID controller and the Arduino programming language. The tester should also be familiar with fluid dynamics and the characters of all the components in this solar boat, like motors, solar panel and so on. The rescuer should be standby to make sure the safety of the pilot.

6. Actual actions during the test

1) Before testing by sea trial, the boat with hydrofoil should be calculated the following things:
   i. Take off velocity and maximum velocity when hydrofoil is working, which contributes to limit the speed when sailing.
   ii. The desired height value and set is as set point in the programming.
   iii. The center of gravity, the center of buoyancy and mechanical balanced point.

2) Before testing by sea trial, the boat with hydrofoil should be made sure this following things:
   i. The center of gravity with pilot is behind the center of buoyancy. When the hydrofoil is not working, the stem should be buoy higher than the stern. This permits a takeoff with the stem up and a tiny additional positive AOA on the wings.
   ii. The IMU sensor should be installed on the boat, and its front side is perpendicular to the z-axis of the boat.
   iii. The stability subsystem can work well.
3) Prepare the battery of 12V and 5V and test it on the day with sunniness.
4) Charge the actuators and Arduino board separately.
5) Take off the boat manually. Keep the power low to limit speed.
6) Turn on the power of Arduino.
7) If the bow cannot reach the desired height when taking off, change the value of proportional action on the height loop in the programming.
8) Disconnect all the batteries and connect the Arduino board with computer.
9) Upload the new program to the Arduino board and then repeat step 5-9 until the bow can reach the desired height
10) If the bow keeps at desired height with oscillation, change the value of integral action on the height loop in the programming.
11) Disconnect all the batteries and connect the Arduino board with computer.
12) Upload the new program to the Arduino board and then repeat step 5-12 until the bow keeps at desired height without oscillation.
13) After taking off, if the stern cannot keep at desired height when sailing, change the value of proportional action on the angle loop in the programming.
14) Disconnect all the batteries and connect the Arduino board with computer.
15) Upload the new program to the Arduino board and then repeat step 5-15 until the stern can keep at desired height after taking off.
16) If the stern keeps at constant with oscillation when sailing, change the value of differential action on the angle loop in the programming.
17) Disconnect all the batteries and connect the Arduino board with computer.
18) Upload the new program to the Arduino board and then repeat step 5-18 until the stern keeps at constant without oscillation when sailing
19) Disconnect all the batteries and connect the Arduino board with computer.
20) Adjust the value of proportional and differential action on velocity loop in programming
21) Upload the new program to the Arduino board and then repeat step 5-21 to eliminate the oscillation.
22) Land the boat on the water.
23) Take off the boat manually and sail it at maximum speed.
24) Repeat step 6- 22. Change the tuning values in programming by trial and error to optimize the steadiness of the control system.
25) Land the boat on the water.
26) Charge the actuators and Arduino board with solar supply.
27) Repeat step 23-25.
28) Get the boat with hydrofoil out of water and record all the turning values.

7. Average and bandwidth

Best value: The hydrofoil can keep the bow at constant height when the boat is sailing. When the boat is taking off, the hydrofoil let the bow of the boat reach the desired height first, and then, rise the stern up to realize level sailing. Additionally, when the hull falls on the water, the actuators of hydrofoil will not move any more.

Bad value: The boat cannot keep at constant height all the time.

Worst value: The boat falls on the water and breaks while testing.

8. Predicted conclusions

Desired case

The desired case is that the hydrofoil can keep the boat at constant height when it is sailing.

Other cases

The possible problems in the process of implementing the height control system were listed as follow. Besides, the corresponding possible reasons which will be helpful to solve this kind of problems were included.

Case 1: No power supply:

- Change the battery.

Case 2: The hydrofoil cannot keep the boat out of water at constant height:

- The tuning value of proportional action needs to be changed.

Case 3: The hydrofoil can keep the boat out of water at constant height only for a short time.

- The tuning value of differential action on angle loop needs to be changed.
- The tuning values of proportional and integral action on velocity loop need to be changed.

Case 4: The boat cannot keep stable all the time

- The power supply from solar panel is not steady.

Appendix I： Test plan of the whole system

Test the whole system of hydrofoil when it is sailing.

1. Introduction

This test plan was written as a reference to test the whole system in the hydrofoil. This whole system only needs to be tested when the boat is sailing out of water.

2. Aim

The aim of this test plan is to prove that the code of the whole system can work well. This code need to be proved that it can realize the function of hydrofoil. The requirements were listed as follow.

- The hydrofoil can keep the boat stable when taking off.
- The hydrofoil can keep the boat stable when sailing.
- The hydrofoil can find the balanced point of the boat again after turning around.
- The hydrofoil can reach the bow at desired height when taking off.
- The hydrofoil can keep the hull of boat at constant height, which also realizes level sailing.

3. Functionality to be tested

- The control system will automatically shut down when the boat falls on the water.
- The hydrofoil can keep the boat stable when taking off.
- The hydrofoil can keep the boat stable when sailing.
- The hydrofoil can find the balanced point of the boat again after turning around.
- The hydrofoil can reach the bow at desired height when taking off.
- The hydrofoil can keep the hull of boat at constant height, which also realizes level sailing.
- The hydrofoil can maintain the hull of the boat at constant height at unstable situation.

4. Hypotheses

Assuming that all the functions of this subsystem work well, then the hypothesis should be: When the boat is sailing, the pilot takes off the hydrofoil manually. The hydrofoil can reach the bow at desired height when taking off. (Shown in Fig 4.1B) After taking off, the boat keep stable and main constant height. (Shown in Fig 4.1C)

5. Target group

The tester should have basic knowledge of PID controller and the Arduino programming language. The tester should also be familiar with fluid dynamics and the characters of all the components in this solar boat, like motors, solar panel and so on. The rescuer should be standby to make sure the safety of the pilot.

6. Actual actions during the test

1) Before testing by sea trial, the boat with hydrofoil should be calculated the following things:

i. Take off velocity and maximum velocity when hydrofoil is working, which contributes to limit the speed when sailing.
ii. The desired height value and set is as set point in the programming.
iii. The center of gravity, the center of buoyancy and mechanical balanced point.
iv. The desired roll and pitch value and set these as set point in the programming.

2) Before testing by sea trial, the boat with hydrofoil should be made sure this following things:
   i. The center of gravity with pilot is behind the center of buoyancy. When the hydrofoil is not working, the stem should be buoy higher than the stern. This permits a takeoff with the stem up and a tiny additional positive AOA on the wings.
   ii. The IMU sensor should be installed on the boat, and its front side is perpendicular to the z-axis of the boat.
3) Search the weather forecast and test it on the day with sunniness.
4) Charge the actuators and Arduino board separately through solar power supply.
5) Take off the boat manually. Keep the power low to limit speed.
6) Turn on the power of Arduino.
7) Observe the situation of the boat and record all the tuning values and take off velocity.
8) After taking off, observe the situation of the boat and record all the tuning values and sailing velocity
9) The pilot controls the boat to turn around at 30 degree. The tester observes the situation of the boat and records it.
10) The pilot controls the boat to turn around at 60 degree. The tester observes the situation of the boat and records it.
11) The pilot controls the boat to turn around at 90 degree. The tester observes the situation of the boat and records it.
12) Land the boat on the water.
13) Disconnect all the power supply and connect the Arduino board with computer.
14) According to the record chart of the situation observed while testing, adjust all the tuning values by trial and error.
15) Upload the new program to the Arduino board and then repeat step 4-15 until the hydrofoil can work well when taking off, sailing and turning around.
16) Get the boat with hydrofoil out of water and record all the turning values.

7. Average and bandwidth

Best value: The hydrofoil can keep the bow at constant height when the boat is sailing. When the boat is taking off, the hydrofoil let the bow of the boat reach the desired height first, and then, rise the stern up to realize level sailing. Additionally, when the hull falls on the water, the actuators of hydrofoil will not move any more.

Bad value: The boat cannot keep at constant height all the time.

Worst value: The boat falls on the water and breaks while testing.

8. Predicted conclusions

Desired case

The desired case is that the when the boat is sailing, the pilot takes off the hydrofoil manually. The hydrofoil can reach the bow at desired height when taking off. (Shown in Fig 4.1B) After taking off, the boat keep stable and main constant height. (Shown in Fig 4.1C)

Other cases

The possible problems in the process of implementing the height control system were listed as follow. Besides, the corresponding possible reasons which will be helpful to solve this kind of problems were included.

Case 1: The boat cannot keep stable all the time

- The power supply from solar panel is not steady.

Case 2: The hydrofoil cannot keep the boat stable.

- Repeat the test plan of stability subsystem. Find out the right tuning values by trial and error method.

Case 3: The hydrofoil cannot keep the hull of boat at constant height.

- Repeat the test plan of stability subsystem. Find out the right tuning values by trial and error method.

Appendix J: The stability subsystem tested in test model

Test the two – wheel car

1. Introduction

This test plan was written as a guide to test the stability subsystem testing in the test model. The test model here is two-wheel car.

2. Aim

The tester listed a list of requirement for each system. The aim of this test plan is to prove that the code of the stability subsystem can make the similar hydrofoil unstable system work well. This similar hydrofoil control system is the two-wheel car. This code need to be proved that it can realize in the dynamic stabilization. The requirements were listed as follow.

- This two-wheel car can realize the dynamic stabilization.
- The control system will automatically shut down when the car falls on the ground

3. Functionality to be tested

- This two-wheel car can stabilize on the ground by itself.
- This two-wheel car can stabilize by itself at unstable situation.
- A sudden force will not lead the car to fall on the ground.
- The control system will automatically shut down when the car falls on the ground.

4. Hypotheses

Assuming that all the functions of this subsystem work well, then the hypothesis should be: The two-wheel car will be stabilized itself when placing it on the ground. And obstacles and a sudden force will not lead the car to fall on the ground. Additionally, when the car falls on the ground, the actuators will not move anymore.

5. Target group

The student who programed this code and built this two-wheel car will be the tester. The person should have basic knowledge of PID controller and the Arduino programming language.

6. Actual actions during the test

1) Find the mechanical balanced point of the two-wheel car and got its balanced roll value to the computer.
2) Get this point 10 times and calculate its average.
3) Use the average as balanced point and change this value in the programming.
4) Prepare the battery of 12V and make sure that the battery is full charged.
5) Charge the Arduino with batteries.
6) Put the car on the ground and push the button to turn on the Arduino board.
7) If the car cannot stand up-right on the ground, change the value of proportional action on

the angle loop in the programming.

8) Disconnect the battery with the Arduino board and connect the Arduino board with computer.
9) Upload the new program to the Arduino board and then repeat step 4-6 until the car can stand up-right on the ground.
10) If the car stands up-right on the ground with oscillation, change the value of differential action on the angle loop in the programming.
11) Disconnect the battery with the Arduino board and connect the Arduino board with computer.
12) Upload the new program to the Arduino board and then repeat step 4-9 until the car can stand on the ground steadily.
13) Push the car to see if the car can still find its balanced point and stay up-right on the ground.
14) If the car falls down, change the value of proportional and integral action on the velocity loop in the programming by trial and error.
15) Disconnect the battery with the Arduino board and connect the Arduino board with computer.
16) Upload the new program to the Arduino board and then repeat step 2, 3, 11 until the car can find its balanced point and stay up-right on the ground.

7. Average and bandwidth

Best value: The car can realize the dynamic stabilization.

Bad value: The car cannot keep stable all the time.

Worst value: The cat cannot even stand on the ground.

8. Predicted conclusions

Desired case

The desired case is that this two-wheel car can work as the hypothesis. All the functions can work well.

Other cases

The possible problems in the process of implementing the stability control system were listed as follow. Besides, the corresponding possible reasons which will be helpful to solve this kind of problems were included.

Case 1: No power supply:

- Change the battery.

Case 2: The car cannot keep stable on the ground

- The tuning value of P action needs to be changed.

Case 3: The car can keep stable on the ground only for a short time.

- The tuning value of D action needs to be changed.

Case 4: The control system of the car cannot turn off when if falls on the ground.

- The angle limitation in the programming needs to be changed.

Appendix K: Code of the stability subsystem

```
// Stability code for hydrofoil
//Date:February to July 2016

//header file
#include <PinChangeInt.h>
#include <Wire.h>
#include "Kalman.h"
#include "I2C.h"

//Create the Kalman instances
Kalman kalmanX;
Kalman kalmanY;

/* IMU Data */
double accX, accY, accZ;// data got from accelerometer
double gyroX, gyroY, gyroZ;// data got from gyroscope
int16_t tempRaw;

double gyroXangle, gyroYangle; // Angle calculated from gyroscope only
double kalAngleX, kalAngleY; // Calculated angle using a Kalman filter

uint32_t timer;
uint8_t i2cData[14]; // Buffer for I2C data

//define motor
#define pwmL 6 //M1 ENA
#define pwmR 5 //M2 ENB
#define MotorL2 7 // IN1
#define MotorL1 8 // IN2
#define MotorR2 4 // IN3
#define MotorR1 3 // IN4

//define encoder
#define hallinterruptR 11 // interrupt hallsensorA INT2
#define hallsensorR    12 // CNT2
#define hallinterruptL 10 //interrupt hall sensorB INT1
#define hallsensorL    9  // CNT1

double KA_P,KA_D; // tuning vales for angle
double KP_P,KP_I;// tuning values for velocity
double KBase;//mechnical balanced pointint SpeedL;
int SpeedR;
int SpeedL;

int pwm,pwml,pwmr;
double AngleBoat,GyroBoat;
int PositionAdd;

void setup(){
  Serial.begin(9600);
  Initial(); // initialize
   Wire.begin();
  TWBR = ((F_CPU / 400000L) - 16) / 2; // Set I2C frequency to 400kHz

  i2cData[0] = 7; // Set the sample rate to 1000Hz - 8kHz/(7+1) = 1000Hz
  i2cData[1] = 0x00; // Disable FSYNC and set 260 Hz Acc filtering, 256 Hz Gyro filtering, 8 KHz sampling
  i2cData[2] = 0x00; // Set Gyro Full Scale Range to ±250deg/s
  i2cData[3] = 0x00; // Set Accelerometer Full Scale Range to ±2g

  while (i2cWrite(0x19, i2cData, 4, false)); // Write to all four registers at once
  while (i2cWrite(0x6B, 0x01, true)); // PLL with X axis gyroscope reference and disable sleep mode
```

```
  while (i2cRead(0x75, i2cData, 1));
  if (i2cData[0] != 0x68) { // Read "WHO_AM_I" register
    Serial.print(F("Error reading sensor"));
    while (1);
  }
 delay(200); // Wait for sensor to stabilize
 /* Set kalman and gyro starting angle */
  while (i2cRead(0x3B, i2cData, 6));
  accX = (i2cData[0] << 8) | i2cData[1];
  accY = (i2cData[2] << 8) | i2cData[3];
  accZ = (i2cData[4] << 8) | i2cData[5];

#ifdef RESTRICT_PITCH
  double roll  = atan2(accY, accZ) * RAD_TO_DEG;
  double pitch = atan(-accX / sqrt(accY * accY + accZ * accZ)) * RAD_TO_DEG;
#else // Eq. 28 and 29
  double roll  = atan(accY / sqrt(accX * accX + accZ * accZ)) * RAD_TO_DEG;
  double pitch = atan2(-accX, accZ) * RAD_TO_DEG;
#endif

  kalmanX.setAngle(roll); // Set starting angle
  kalmanY.setAngle(pitch);
  gyroXangle = roll;
  gyroYangle = pitch;

 PCintPort::attachInterrupt(hallinterruptL, EncoderL,FALLING);
 PCintPort::attachInterrupt(hallinterruptR, EncoderR,FALLING);
  }

void loop(){
  double AngleBoatAvg[3];
  double AngleAvg=0;
 AngleBoatAvg[0]=0;
 AngleBoatAvg[1]=0;
 AngleBoatAvg[2]=0;
  while (1){
    if(UpdateAttitude()){

      AngleBoatAvg[2] = AngleBoatAvg[1];
      AngleBoatAvg[1] = AngleBoatAvg[0];
       AngleBoatAvg[0] = AngleBoat;
       AngleAvg= (AngleBoatAvg[0]+AngleBoatAvg[1]+AngleBoatAvg[2])/3;
       if (AngleAvg <40 || AngleAvg>-40 ){
         MotorSpeedOutput();
         Deadtime();
         }
       }
    }
  }

//calculate pwm
void MotorSpeedOutput(){
  float MotorSpeed;
  //error
  MotorSpeed=(SpeedL+SpeedR)*0.5;// two motor pulse average
  SpeedL=0;
  SpeedR=0;
  //Accumulation
  PositionAdd += MotorSpeed;
  PositionAdd = constrain(MotorSpeed,-800,800);

  pwm = (AngleBoat - KBase) * KA_P // KA_P: P action on Angle
       + GyroBoat * KA_D          // KA_D: D action on angular velocity
       + MotorSpeed*KP_P        //KP_P: P action on veloctiy
       + PositionAdd *KP_I;  //KP_I: I action on position
  }
```

```
  //initialize
  void Initial(){
  pinMode(hallsensorL, INPUT);
  pinMode(hallsensorR, INPUT);
  pinMode(pwmL, OUTPUT);
  pinMode(pwmR, OUTPUT);
  pinMode(MotorL1, OUTPUT);
  pinMode(MotorL2, OUTPUT);
  pinMode(MotorR1, OUTPUT);
  pinMode(MotorR2, OUTPUT);
// tuning values should be found out later
  KA_P =1;
  KA_D =1;
  KP_P =1;
  KP_I =1;
  KBase =1;

  SpeedL = 0;
  SpeedR = 0;
  pwm=0,pwml=0,pwmr=0;
    }
 void Deadtime() //Deadtime compensation && saturate
{

 if (pwml<0)
  {
    digitalWrite(MotorL1, HIGH);
    digitalWrite(MotorL2, LOW);
    pwml =- pwml;
  }
  else
  {
    digitalWrite(MotorL1, LOW);
    digitalWrite(MotorL2, HIGH);
  }

  if (pwmr<0)
  {
    digitalWrite(MotorR1, LOW);
    digitalWrite(MotorR2, HIGH);
    pwmr = -pwmr;
  }
  else
  {
   digitalWrite(MotorR1, HIGH);
   digitalWrite(MotorR2, LOW);
  }
  if( AngleBoat > 45 || AngleBoat < -45 )
  {
    pwml = 0;
    pwmr = 0;
  }
  //saturate
  if(pwml > 255)
 { pwml = 255; }
  if(pwmr > 255)
  { pwmr = 255; }
  analogWrite(pwmR,pwmr);
  analogWrite(pwmL, pwml);

}
//encoder pluse
void EncoderL()
{
  if (digitalRead(hallsensorR))
    SpeedL += 1;
  else
    SpeedL -= 1;

}
```

```
void EncoderR()
{
  if (!digitalRead(hallsensorL))//same wheel direaction but different motor direaction
    SpeedR += 1;
  else
    SpeedR -= 1;
   }

   int UpdateAttitude()
{
 if((micros() - timer) >= 10000)
 {    //10ms
  //Update all the values
  while (i2cRead(0x3B, i2cData, 14));
  accX = ((i2cData[0] << 8) | i2cData[1]);
  accY = ((i2cData[2] << 8) | i2cData[3]);
  accZ = ((i2cData[4] << 8) | i2cData[5]);
  tempRaw = (i2cData[6] << 8) | i2cData[7];
  gyroX = (i2cData[8] << 8) | i2cData[9];
  gyroY = (i2cData[10] << 8) | i2cData[11];
  gyroZ = (i2cData[12] << 8) | i2cData[13];

  double dt = (double)(micros() - timer) / 1000000; // Calculate delta time
  timer = micros();

  // atan2 outputs the value of -π to π (radians)
  // It is then converted from radians to degrees
#ifdef RESTRICT_PITCH
  double roll  = atan2(accY, accZ) * RAD_TO_DEG;
  double pitch = atan(-accX / sqrt(accY * accY + accZ * accZ)) * RAD_TO_DEG;
#else // Eq. 28 and 29
  double roll  = atan(accY / sqrt(accX * accX + accZ * accZ)) * RAD_TO_DEG;
  double pitch = atan2(-accX, accZ) * RAD_TO_DEG;
#endif

  double gyroXrate = gyroX / 131.0; // Convert to deg/s
  double gyroYrate = gyroY / 131.0; // Convert to deg/s

#ifdef RESTRICT_PITCH
  // This fixes the transition problem when the accelerometer angle jumps between -180 and 180 degrees
  if ((roll < -90 && kalAngleX > 90) || (roll > 90 && kalAngleX < -90)) {
    kalmanX.setAngle(roll);
    kalAngleX = roll;

    gyroXangle = roll;
  } else
    kalAngleX = kalmanX.getAngle(roll, gyroXrate, dt); // Calculate the angle using a Kalman filter

  if (abs(kalAngleX) > 90)
    gyroYrate = -gyroYrate;
  kalAngleY = kalmanY.getAngle(pitch, gyroYrate, dt);
#else
  // This fixes the transition problem when the accelerometer angle jumps between -180 and 180 degrees
  if ((pitch < -90 && kalAngleY > 90) || (pitch > 90 && kalAngleY < -90)) {
    kalmanY.setAngle(pitch);

    kalAngleY = pitch;
    gyroYangle = pitch;
  } else
    kalAngleY = kalmanY.getAngle(pitch, gyroYrate, dt); // Calculate the angle using a Kalman filter

  if (abs(kalAngleY) > 90)
    gyroXrate = -gyroXrate; // Invert rate, so it fits the restriced accelerometer reading
  kalAngleX = kalmanX.getAngle(roll, gyroXrate, dt); // Calculate the angle using a Kalman filter
#endif

  AngleBoat = kalAngleX;
  GyroBoat = gyroXrate;
   return 1;
 }
  return 0;
}
```

Appendix L: Code of the height subsystem

```
// heightcontroller
//Date:February to July 2016

//header file
#include <PinChangeInt.h>
#include <Wire.h>
#include "Kalman.h"
#include "I2C.h"

//Create the Kalman instances
Kalman kalmanX;
Kalman kalmanY;

/* IMU Data */
double accX, accY, accZ;
double gyroX, gyroY, gyroZ;
int16_t tempRaw;

double gyroXangle, gyroYangle; // Angle calculate using the gyro only
double kalAngleX, kalAngleY; // Calculated angle using a Kalman filter

uint32_t timer;
uint8_t i2cData[14]; // Buffer for I2C data

//define motor
#define pwmF 6 //M1 ENA front foil
#define pwmB 5 //M2 ENB Back foil
#define MotorF1 7 // IN1
#define MotorF2 8 // IN2
#define MotorB2 4 // IN3
#define MotorB1 3 // IN4

//define ultrasonic
int trig = 10;    //
int echo = 9;    //

// Can be changed later
const int numReadings = 10;// buffer for waves
int setpoint = 40;

int readings[numReadings];      // the readings from the analog input
int readIndex = 0;              // the index of the current reading
```

```
int total = 0;                  // the running total
int average = 0;                // the average

//define encoder
#define hallinterruptF 11 // interrupt hallsensorA INT2
#define hallsensorF    12 // CNT2
#define hallinterruptB 10 //interrupt hall sensorB INT1
#define hallsensorB    9  // CNT1

int SpeedF;
int SpeedB;
int pwm,pwmf,pwmb;
float HeightSpeed;
double AngleHeight,GyroHeight;
double KA_P,KA_D;
double KP_P,KP_I;
double KBase;
double KH_P,KH_I,KH_D;
int PositionAdd;
int E_0, E_1;


void setup(){
  Serial.begin(9600);
  Initial(); // initialize the module
  for (int thisReading = 0; thisReading < numReadings; thisReading++) {
    readings[thisReading] = 0;
    }
   Wire.begin();
  TWBR = ((F_CPU / 400000L) - 16) / 2; // Set I2C frequency to 400kHz

  i2cData[0] = 7; // Set the sample rate to 1000Hz - 8kHz/(7+1) = 1000Hz
  i2cData[1] = 0x00; // Disable FSYNC and set 260 Hz Acc filtering, 256 Hz Gyro filtering, 8 KHz sampling
  i2cData[2] = 0x00; // Set Gyro Full Scale Range to ±250deg/s
  i2cData[3] = 0x00; // Set Accelerometer Full Scale Range to ±2g

  while (i2cWrite(0x19, i2cData, 4, false)); // Write to all four registers at once
  while (i2cWrite(0x6B, 0x01, true)); // PLL with X axis gyroscope reference and disable sleep mode

  while (i2cRead(0x75, i2cData, 1));
  if (i2cData[0] != 0x68) { // Read "WHO_AM_I" register
    Serial.print(F("Error reading sensor"));
    while (1);
  }
 delay(200); // Wait for sensor to stabilize
 /* Set kalman and gyro starting angle */
  while (i2cRead(0x3B, i2cData, 6));
  accX = (i2cData[0] << 8) | i2cData[1];
  accY = (i2cData[2] << 8) | i2cData[3];
  accZ = (i2cData[4] << 8) | i2cData[5];

#ifdef RESTRICT_PITCH
  double roll  = atan2(accY, accZ) * RAD_TO_DEG;
  double pitch = atan(-accX / sqrt(accY * accY + accZ * accZ)) * RAD_TO_DEG;
#else // Eq. 28 and 29
  double roll  = atan(accY / sqrt(accX * accX + accZ * accZ)) * RAD_TO_DEG;
  double pitch = atan2(-accX, accZ) * RAD_TO_DEG;
#endif

  kalmanX.setAngle(roll); // Set starting angle
  kalmanY.setAngle(pitch);
  gyroXangle = roll;
  gyroYangle = pitch;
```

```
  PCintPort::attachInterrupt(hallinterruptF, EncoderF,FALLING);
  PCintPort::attachInterrupt(hallinterruptB, EncoderB,FALLING);
  }

void loop(){
  double AnglePitchAvg[3];
  double AngleAvg=0;
  AnglePitchAvg[0]=0;
  AnglePitchAvg[1]=0;
  AnglePitchAvg[2]=0;
  while (1){
    if(UpdateAttitude()){

      AnglePitchAvg[2] = AnglePitchAvg[1];
      AnglePitchAvg[1] = AnglePitchAvg[0];
      AnglePitchAvg[0] = AngleHeight;
      AngleAvg= (AnglePitchAvg[0]+AnglePitchAvg[1]+AnglePitchAvg[2])/3;
      if (AngleAvg <40 || AngleAvg>-40 ){
        MotorSpeedOutput();
        Deadtime();
        }
      }
    }
  }
```

```
//calculate pwm
void MotorSpeedOutput(){
  //error
  HeightSpeed=(SpeedF+SpeedB)*0.5;// two motor pulse average
  SpeedF=0;
  SpeedB=0;
  //Accumulation
  PositionAdd += HeightSpeed;
  PositionAdd = constrain(HeightSpeed,-800,800);//这个800
   long IntervalTime=0; //
  while(1){
    digitalWrite(trig, 1);//
    delayMicroseconds(15);//
    digitalWrite(trig, 0);
    IntervalTime=pulseIn(echo, HIGH);
    float S=IntervalTime/58.00;
   total = total - readings[readIndex];
  // read from the sensor:
  readings[readIndex] = S;

 // add the reading to the total:
  total = total + readings[readIndex];
  // advance to the next position in the array:
  readIndex = readIndex + 1;
  Serial.print("Newdata=");
 Serial.println(readings[readIndex]);
  // if we're at the end of the array...
  if (readIndex >= numReadings) {
    // ...wrap around to the beginning:
    readIndex = 0;
    }

  // calculate the average:
  average = total / numReadings;
  Serial.print("Average=");
  Serial.println(average);
  delay(100);        // delay in between reads for stability
  }

  pwmf =KH_P*E_0 + KH_D*(E_0-E_1);  // only calculated from ultransonic sensor

  pwmb =  (AngleHeight-KBase)*KA_P //p
       + GyroHeight * KA_D          //D
       + HeightSpeed*KP_I        //I
       + PositionAdd *KP_P;  //P
  }
  //initialize
  void Initial(){
  pinMode(hallsensorF, INPUT);
  pinMode(hallsensorB, INPUT);
  pinMode(pwmF, OUTPUT);
  pinMode(pwmB, OUTPUT);
  pinMode(MotorF1, OUTPUT);
  pinMode(MotorF2, OUTPUT);
  pinMode(MotorB1, OUTPUT);
  pinMode(MotorB2, OUTPUT);
     pinMode(echo, INPUT);
  pinMode(trig, OUTPUT);

  SpeedF = 0;
  SpeedB = 0;
 pwmf=0,pwmb=0;
    }
```

```
void Deadtime() //Deadtime compensation && saturate
{

 if (pwmf<0)
  {
    digitalWrite(MotorF1, HIGH);
    digitalWrite(MotorF2, LOW);
    pwmf =- pwmf;
  }
  else
  {
    digitalWrite(MotorF1, LOW);
    digitalWrite(MotorF2, HIGH);
  }

  if (pwmb<0)
  {
    digitalWrite(MotorB1, LOW);
    digitalWrite(MotorB2, HIGH);
    pwmb = -pwmb;
  }
  else
  {
    digitalWrite(MotorB1, HIGH);
    digitalWrite(MotorB2, LOW);
  }
  if( AngleHeight > 45 || AngleHeight < -45 )
  {
    pwmf = 0;
    pwmb = 0;
  }
  //saturate
  if(pwmf > 255)
 { pwmf = 255; }
  if(pwmb > 255)
  { pwmb = 255; }
  analogWrite(pwmF,pwmf);
  analogWrite(pwmB, pwmb);
}

//encoder pluse
void EncoderF()

 {
   if (digitalRead(hallsensorF))
     SpeedF += 1;
   else
     SpeedF -= 1;
     }

void EncoderB()
 {
   if (digitalRead(hallsensorB))//same wheel direaction but different motor direaction
     SpeedB += 1;
   else
     SpeedB -= 1;
    }



    int UpdateAttitude()
 {
```

```
 if((micros() - timer) >= 10000)
 {    //10ms
  //Update all the values
  while (i2cRead(0x3B, i2cData, 14));
  accX = ((i2cData[0] << 8) | i2cData[1]);
  accY = ((i2cData[2] << 8) | i2cData[3]);
  accZ = ((i2cData[4] << 8) | i2cData[5]);
  tempRaw = (i2cData[6] << 8) | i2cData[7];
  gyroX = (i2cData[8] << 8) | i2cData[9];
  gyroY = (i2cData[10] << 8) | i2cData[11];
  gyroZ = (i2cData[12] << 8) | i2cData[13];

  double dt = (double)(micros() - timer) / 1000000; // Calculate delta time
  timer = micros();

  // atan2 outputs the value of -π to π (radians)
  // It is then converted from radians to degrees
#ifdef RESTRICT_PITCH
  double roll  = atan2(accY, accZ) * RAD_TO_DEG;
  double pitch = atan(-accX / sqrt(accY * accY + accZ * accZ)) * RAD_TO_DEG;
#else // Eq. 28 and 29
  double roll  = atan(accY / sqrt(accX * accX + accZ * accZ)) * RAD_TO_DEG;
  double pitch = atan2(-accX, accZ) * RAD_TO_DEG;
#endif

  double gyroXrate = gyroX / 131.0; // Convert to deg/s
  double gyroYrate = gyroY / 131.0; // Convert to deg/s

#ifdef RESTRICT_PITCH
  // This fixes the transition problem when the accelerometer angle jumps between -180 and 180 degrees
  if ((roll < -90 && kalAngleX > 90) || (roll > 90 && kalAngleX < -90)) {
    kalmanX.setAngle(roll);
    kalAngleX = roll;
    gyroXangle = roll;
  } else
    kalAngleX = kalmanX.getAngle(roll, gyroXrate, dt); // Calculate the angle using a Kalman filter

  if (abs(kalAngleX) > 90)
    gyroYrate = -gyroYrate;
  kalAngleY = kalmanY.getAngle(pitch, gyroYrate, dt);
#else
  // This fixes the transition problem when the accelerometer angle jumps between -180 and 180 degrees
  if ((pitch < -90 && kalAngleY > 90) || (pitch > 90 && kalAngleY < -90)) {
    kalmanY.setAngle(pitch);

    kalAngleY = pitch;
    gyroYangle = pitch;
  } else
    kalAngleY = kalmanY.getAngle(pitch, gyroYrate, dt); // Calculate the angle using a Kalman filter

  if (abs(kalAngleY) > 90)
    gyroXrate = -gyroXrate; // Invert rate, so it fits the restriced accelerometer reading
  kalAngleX = kalmanX.getAngle(roll, gyroXrate, dt); // Calculate the angle using a Kalman filter
#endif

  AngleHeight = kalAngleX;
  GyroHeight = gyroXrate;
  E_0=average-setpoint;
  E_1=E_0;

   return 1;
 }
  return 0;
}
```

Appendix M: Code of angular velocity at x-axis

```
#include <Wire.h>
#include "ComPacket.h"
#include "I2C.h"

/* IMU Data */
double gyroX;


double gyroXangle; // Angle calculate using the gyro only
double GyroHydrofoil;

uint32_t timer;
uint8_t i2cData[14]; // Buffer for I2C data


void setup() {
  Serial.begin(9600);
  Wire.begin();
  TWBR = ((F_CPU / 400000L) - 16) / 2; // Set I2C frequency to 400kHz

  i2cData[0] = 7; // Set the sample rate to 1000Hz - 8kHz/(7+1) = 1000Hz
  i2cData[1] = 0x00; // Disable FSYNC and set 260 Hz Acc filtering, 256 Hz
  i2cData[2] = 0x00; // Set Gyro Full Scale Range to ±250deg/s
  i2cData[3] = 0x00; // Set Accelerometer Full Scale Range to ±2g

 while (i2cWrite(0x19, i2cData, 4, false));
 while (i2cWrite(0x6B, 0x01, true));

 while (i2cRead(0x75, i2cData, 1));
 if (i2cData[0] != 0x68) { // Read "WHO_AM_I" register
   Serial.print(F("Error reading sensor"));
   while (1);
 }

 delay(200); // Wait for sensor to stabilize

 timer = micros();
 }

void loop() {

 while(1)
   {
     UpdateAttitude();
}
   }
 int UpdateAttitude()
{
 if((micros() - timer) >= 10000)

{    //10ms
 /* Update all the values */
 while (i2cRead(0x3B, i2cData, 14));
 gyroX = (i2cData[8] << 8) | i2cData[9];

 double dt = (double)(micros() - timer) / 1000000; // Calculate delta time
 timer = micros();
```

```
  double gyroXrate = gyroX / 131.0; // Convert to deg/s
  GyroHydrofoil = gyroXrate;
 Serial.print("GyroHydrofoil");Serial.println(GyroHydrofoil);
 return 1;
 }
  return 0;
}
```

Appendix N: Code of angle of inclination at x-axis

```
#include <Wire.h>
#include "ComPacket.h"
#include "I2C.h"
#include "Kalman.h"

Kalman kalmanX;
Kalman kalmanY;

/* IMU Data */
double accX, accY, accZ;
double gyroX, gyroY, gyroZ;


double gyroXangle, gyroYangle; // Angle calculate using the gyro only
double accAngleX, accAngleY; // Calculated angle using a Kalman filter


uint32_t timer;
uint8_t i2cData[14]; // Buffer for I2C data

double AccAngle;
double GyroAngle;
double kalAngle;
void setup() {
  Serial.begin(9600);
  Wire.begin();
  TWBR = ((F_CPU / 400000L) - 16) / 2; // Set I2C frequency to 400kHz

  i2cData[0] = 7; // Set the sample rate to 1000Hz - 8kHz/(7+1) = 1000Hz
  i2cData[1] = 0x00; // Disable FSYNC and set 260 Hz Acc filtering, 256 Hz Gy
  i2cData[2] = 0x00; // Set Gyro Full Scale Range to ±250deg/s
  i2cData[3] = 0x00; // Set Accelerometer Full Scale Range to ±2g

  while (i2cWrite(0x19, i2cData, 4, false)); // Write to all four registers a
  while (i2cWrite(0x6B, 0x01, true)); // PLL with X axis gyroscope reference

  while (i2cRead(0x75, i2cData, 1));
  if (i2cData[0] != 0x68) { // Read "WHO_AM_I" register
    Serial.print(F("Error reading sensor"));
    while (1);
  }

  delay(200); // Wait for sensor to stabilize

  timer = micros();
}

void loop() {
  while(1)
    {
      UpdateAttitude();

      }

    }
```

```
int UpdateAttitude()
{
 if((micros() - timer) >= 10000)
 {    //10ms
  /* Update all the values */
  while (i2cRead(0x3B, i2cData, 14));
  accX = ((i2cData[0] << 8) | i2cData[1]);
  accY = ((i2cData[2] << 8) | i2cData[3]);
  accZ = ((i2cData[4] << 8) | i2cData[5]);

  gyroX = (i2cData[8] << 8) | i2cData[9];
  gyroY = (i2cData[10] << 8) | i2cData[11];
  gyroZ = (i2cData[12] << 8) | i2cData[13];

  double dt = (double)(micros() - timer) / 1000000;
  timer = micros();

  double AccAngle  = atan(accY/accZ) * RAD_TO_DEG;
  double gyroXrate = gyroX / 131.0; // Convert to deg/s
  double kalAngle = kalmanX.getAngle(AccAngle, gyroXrate, dt);
  GyroAngle= GyroAngle +gyroXrate * dt;

Serial.print(GyroAngle); Serial.print(",");
Serial.print(AccAngle);Serial.print(","); Serial.println(kalAngle);


   return 1;
 }
  return 0;
}
```

Appendix O: Code of the height measurement

```
// smoothing ultrasonic sensor
//Date: February to June 2016

const int trig = 10;    //
const int echo = 9;    //
const int numReadings = 10;

int readings[numReadings];      // the readings from the analog input
int readIndex = 0;              // the index of the current reading
int total = 0;                  // the running total
int average = 0;                // the average

void setup() {
  pinMode(echo, INPUT);
  pinMode(trig, OUTPUT);

  Serial.begin(9600);
    for (int thisReading = 0; thisReading < numReadings; thisReading++) {
    readings[thisReading] = 0;
    }
}
void loop() {
  long IntervalTime=0; //
  while(1){
    digitalWrite(trig, 1);//
    delayMicroseconds(15);//
    digitalWrite(trig, 0);
    IntervalTime=pulseIn(echo, HIGH);
    float S=IntervalTime/58.00;
    //Serial.print("S=");
    //Serial.println(S);


  total = total - readings[readIndex];
  // read from the sensor:
  readings[readIndex] = S;


  // add the reading to the total:
  total = total + readings[readIndex];
  // advance to the next position in the array:
  readIndex = readIndex + 1;
  Serial.print("Newdata=");
 Serial.println(readings[readIndex]);
  // if we're at the end of the array...
  if (readIndex >= numReadings) {
    // ...wrap around to the beginning:
    readIndex = 0;
    }

  // calculate the average:
  average = total / numReadings;
  // send it to the computer as ASCII digits

  Serial.print("Average=");
  Serial.println(average);
  delay(100);        // delay in between reads for stability
  }
}
```

Appendix P: Planning

Planning																					
(10/2/2016-29/6/2016)																					
Activities	Month	February				March				April				May				June			
	Week	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Orientation Phase																					
Research proposal																					
Problem definition																					
Theory framework																					
Research method																					
Planning																					
Communicate and improve																					
Exection Phase																					
Self-study PID controller																					
Requirements analysis																					
Whole system design																					
Stability sub system desgin																					
Design the electronic circuit																					
Height sub system design																					
Design the electronic circuit																					
Component design & testing																					
Gather requirements																					
Program with IMU sensor																					
Test the program																					
Optimize the program																					
Program with ultrasonic sensor																					
Test the program																					
Optimize the program																					
Design the tunning values																					
Design the algorithm of PID																					
Coding																					
In-company presentation																					
Build the test model																					
Discussion & Modification																					
Sub system integration testing																					
Test the code with test model																					
Discussion & Modification																					
Optimize the program																					
System integration testing																					
Interface design																					
User acceptance																					
User manual																					
Write report																					
Conclusion Phase																					
Modify report																					
In-company presentation																					
In-school final presentaion																					

Appendix Q: Workflow of improving the compass sensor

In designing and testing, the scientific method will be obeyed. (Shown in fig 3.1) There are four steps in one work flow.

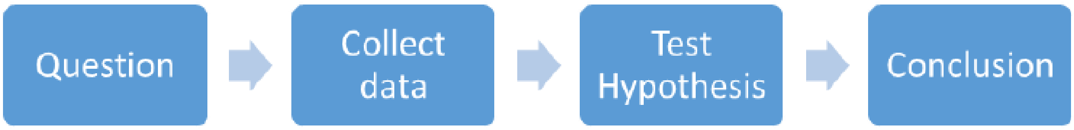
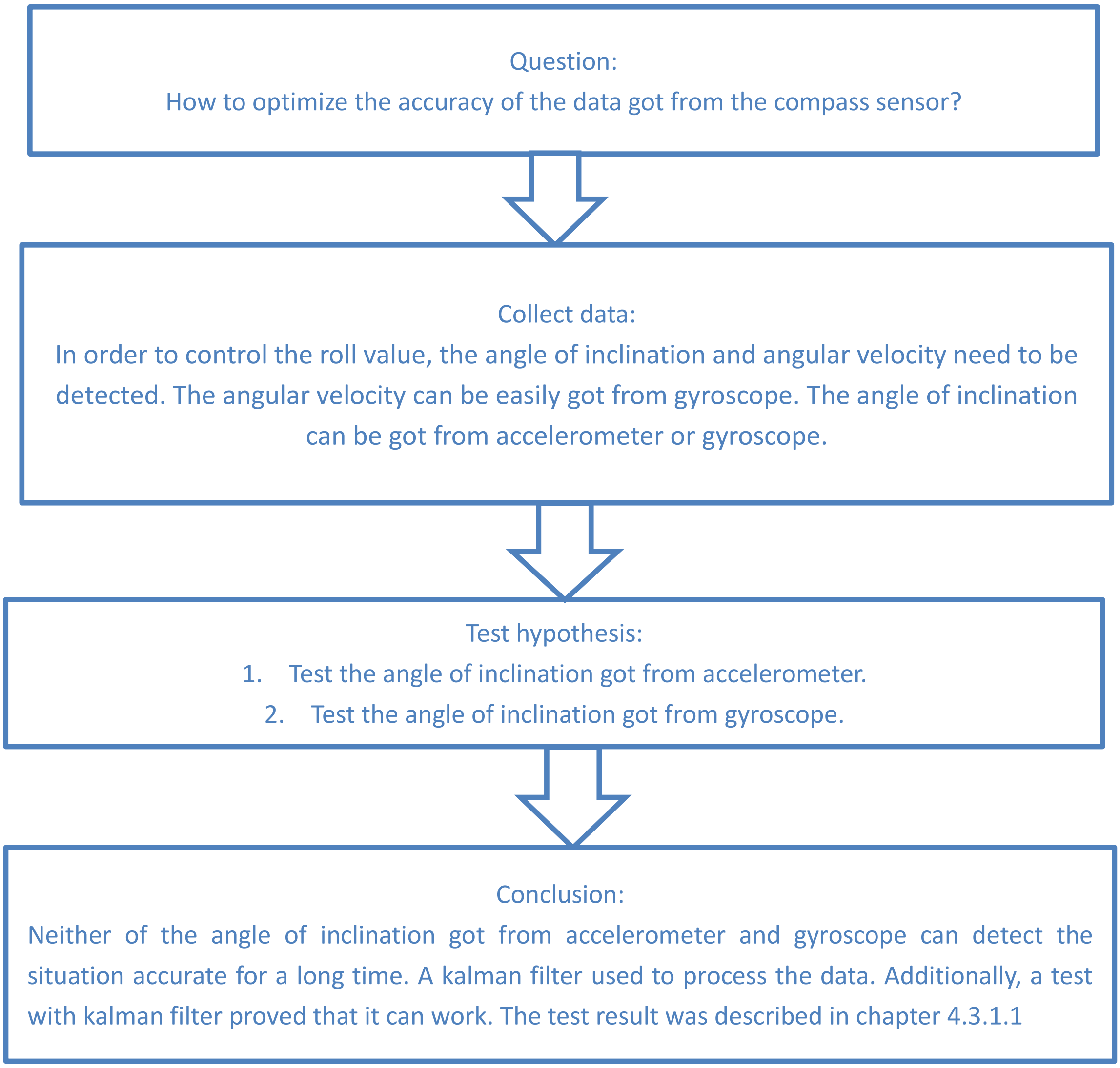


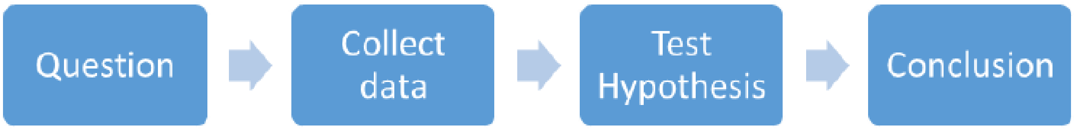
Figure 10 Scientific method

A new work flow will be processed depend on the foundation of the conclusion of last work flow. The work flow will end when the experiment has got a satisfied result. It's also an easier way for other researchers to continue with the experiment improvements.



Appendix R: Workflow of improving the ultrasonic sensor

In designing and testing, the scientific method will be obeyed. (Shown in fig 3.1) There are four steps in one work flow.



A new work flow will be processed depend on the foundation of the conclusion of last work flow. The work flow will end when the experiment has got a satisfied result. It's also an easier way for other researchers to continue with the experiment improvements.

Question:
How to optimize the accuracy of the data got from the ultrasonic sensor?

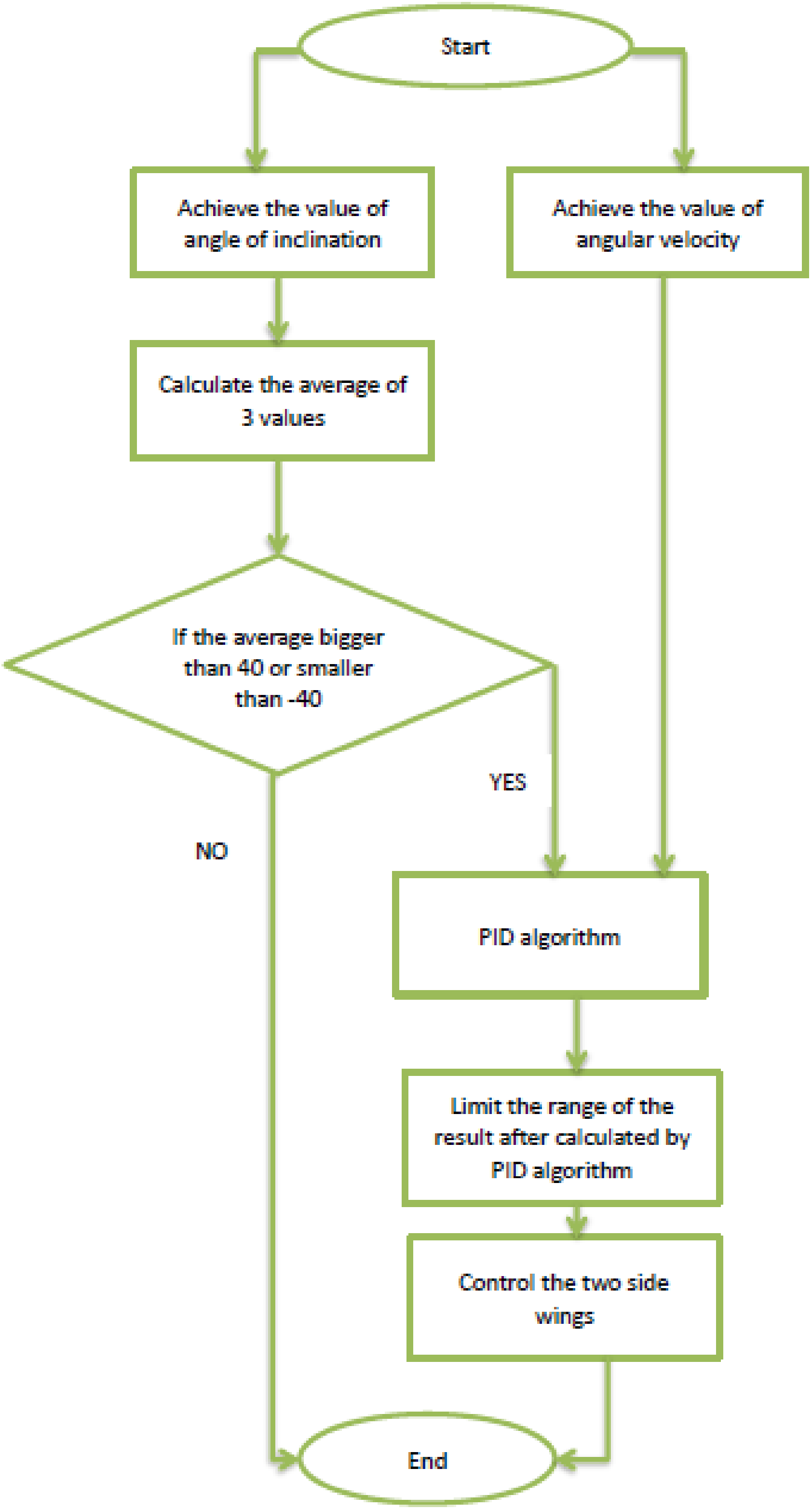
Collect data:
In order to control the height, the influence of the waves needs to be reduced. The data got from ultrasonic sensor tested in practice first.

Test hypothesis:

1. Test the old type ultrasonic senor.
2. Test the new type ultrasonic sensor.
3. Test the ultrasonic sensor with improved program.

Conclusion:
The waterproof problem was solved by selecting a new type. A smoothing function was added in the software to prove that it can work. The test result was described in chapter 4.3.2.1

Appendix S: The flow chart of stability subsystem



Appendix T: The flow chart of height subsystem