Software Requirements Specification

for

RoboSub Controls Subsystem

Version 1.0 approved

Prepared by

Albert Chew

Gabriela Cortes

Sana Shaikh

California State University Department of Engineering

December 12, 2020

**Table of Contents**

Table of Contents.................................................................................................................2

Revision History...................................................................................................................3

1. Introduction......................................................................................................................4

1.1. Purpose..................................................................................................................4

1.2. Intended Audience and Reading Suggestions.......................................................4

1.3. Product Scope......................................................................................................4

1.4. Definitions, Acronyms, and Abbreviations .........................................................5

1.5. References.............................................................................................................5

2. Overall Description..........................................................................................................6

2.1. System Analysis…................................................................................................6

2.2. Product Perspective...............................................................................................6

2.3. Product Functions.................................................................................................7

2.4. User Classes and Characteristics..........................................................................8

2.5. Operating Environment........................................................................................8

2.6. Design and Implementation Constraints...............................................................8

2.7. User Documentation.............................................................................................8

2.8. Assumptions and Dependencies........................................................................10

2.9. Apportioning of Requirements..........................................................................10

3. External Interface Requirements................................................................................11

3.1. User Interfaces...................................................................................................11

3.2. Hardware Interfaces..........................................................................................11

3.3. Software Interfaces...........................................................................................11

3.4. Communications Interfaces...............................................................................11

4. Requirements Specification.......................................................................................12

4.1. Functional Requirements...................................................................................12

4.2. External Interface Requirements.......................................................................12

4.3. Logical Database Requirements.........................................................................12

4.4. Design Constraints..............................................................................................12

5. Other Nonfunctional Requirements.............................................................................13

5.1. Performance Requirements................................................................................13

5.2. Safety Requirements..........................................................................................13

5.3. Security Requirements........................................................................................13

5.4. Software Quality Attributes................................................................................13

5.5. Business Rules....................................................................................................13

6. Legal and Ethical Considerations.…..........................................................................14

Appendix A: Glossary............................................................................................15

Appendix B: Analysis Models...............................................................................16

Appendix C: To Be Determined List.....................................................................16

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Date | Reason For Changes | Version |
| Albert, Gaby, Sana | 12/10/2020 | Delivery of the Software Requirements Document | Draft # 1 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**1. Introduction**

The purpose of the Controls subsystem is to control the Robosub movement with eight thrusters using PID controllers. We used three different axes for control movement. Pitch axis to tilt the sub forward or backwards, Roll axis to move the sub side to side, and YAW to rotate the sub right and left.

**1.1 Purpose**

The purpose of this document is to explain how we will be using PID controllers for horizontal and vertical movement. We will also be providing diagrams to explain how we designed each PID controller for each axis.

**1.2 Intended Audience and Reading Suggestions**

This document is intended for next year's senior design team to understand our goal and continue the development of horizontal and vertical movement.

**1.3 Product Scope**

Arduino IDE is used to create Arduino Sketches and load them onto the microcontroller, so it can control and send the proper outputs to the thrusters of the Robosub.

The microcontroller will help balance, move forward and backward, left and right, and up and down as it receives input from the IMU sensors and SMACH node.

**1.4 Definitions, Acronyms, and Abbreviations**

AUV – Autonomous Underwater Vehicle

IMU – Inertial Measuring Unit

PID – Proportional Integral Derivative

PWM – Pusle Width Modulation

ROS – Robot Operating System

SMACH – State Machine

**1.5 References**

PID Library - <https://github.com/br3ttb/Arduino-PID-Library/blob/master/PID_v1.h>

Setting up a PID on a drone - <https://www.technik-consulting.eu/en/optimizing/drone_PID-optimizing.html>

PID control with Arduino - <http://www.electronoobs.com/eng_robotica_tut6.php>

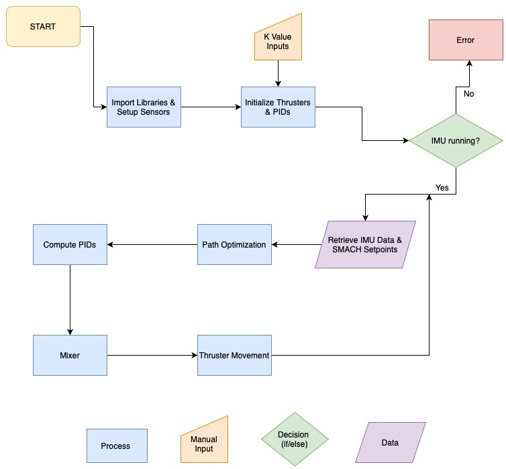
**2. Overall Description**

**2.1 System Analysis**

1. The goal of the Controls subsystem is to implement PID controllers in Arduino IDE to help balance the AUV by programming for the horizontal and vertical thrusters of the Robosub.
2. The major technical hurdle that we may encounter is not being able to test the Arduino code on the Robosub.
3. A possible solution to our problem would be getting access to the Cal State LA’s Robosub lab.

**2.2 Product Perspective**

The code developed by the Controls subsystem is to be used in conjunction with the other subsystems, namely Navigation and Mission Planning. This will help ensure that the subsystems are communicating together in order to make the Robosub balance and move smoothly. The code developed will rely on information sent from the Navigation and Mission Planning subsystems, such as the IMU sensor data, as well as the SMACH node. Fig. 1. shows the flowchart for our prospective code.



*Figure 1 – Code Flowchart*

**2.3 Product Functions**

This section provides a summary of the major functions that the developed code will perform. As stated in section 1, the Controls subsystem’s developed Arduino microcontroller will be responsible for the following:

* Initializing the thrusters and PID controllers
* Running a decision loop to check if the IMU sensor is running
* Retrieve IMU data and SMACH setpoints
* Performing path optimization
* Computing the PWM speed from the PID algorithm
* Combining the different output values together
* Sending the final PWM speed to the eight thrusters

**2.4 User Classes and Characteristics**

The various user classes that may be anticipated to use this product are software developers for next year’s senior design team, or the Controls team members from the AUV club that plan to work on the Controls subsystem using Arduino IDE and PID controllers.

**2.5 Operating Environment**

The environment that the Controls subsystem will be using to develop the code is called the Arduino IDE. We will also be using ROS Melodic on Ubuntu 18.04.

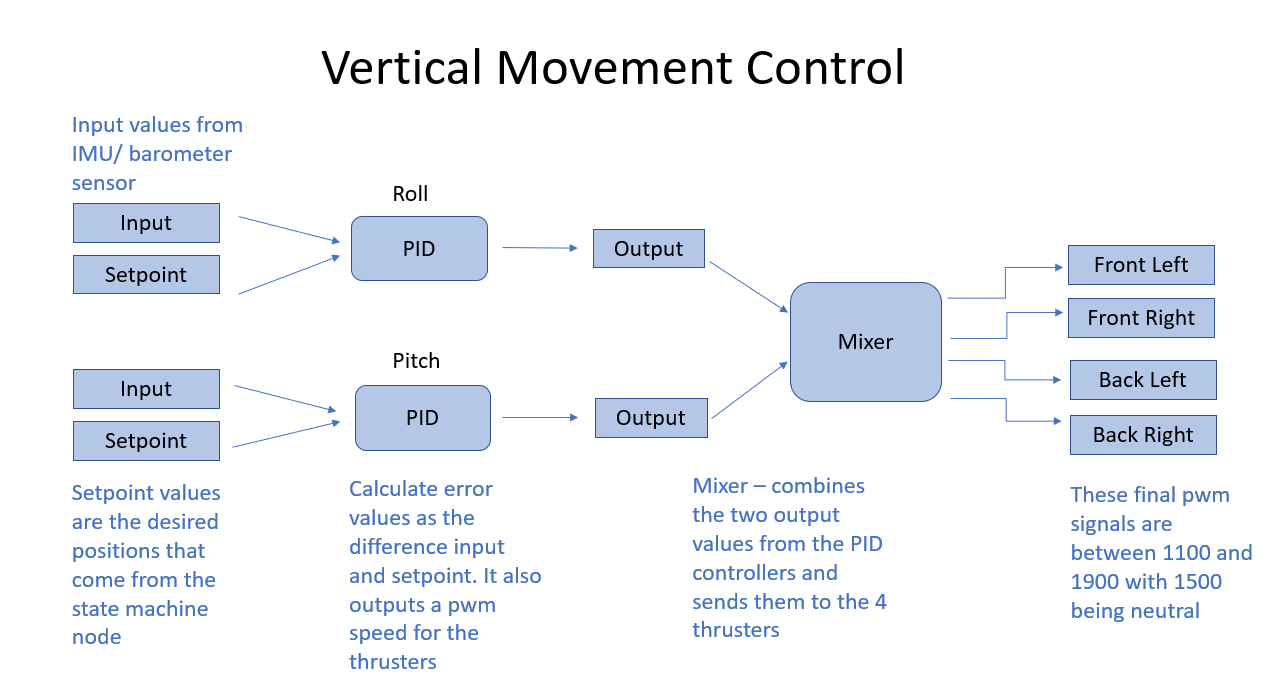
**2.6 Design and Implementation Constraints**

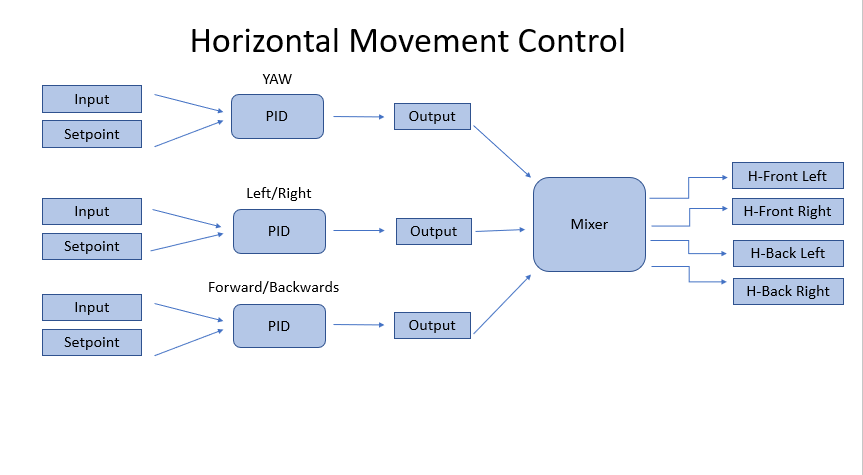
Some of the constraints of the Robosub are as follows:

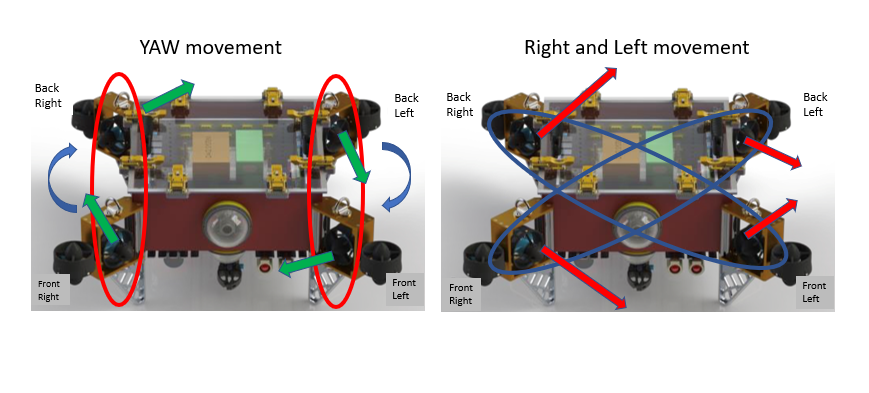
* Communication between the software developers and hardware developers working on the same or different parts of the Robosub.
* Performing different test on the Robosub with the restrictions of the 2020-2021 school year.

**2.7 User Documentation**

We will be providing visual representation of the movements of the RoboSub and block diagrams for our code.







**2.8 Assumptions and Dependencies**

The requirements stated in this document may be affected depending on whether we’ll have access to the lab on campus so the correct K values for the PID controllers could be obtained. In the meantime, we will be using arbitrary K values to test the PID algorithm.

**2.9 Apportioning of Requirements**

Our Senior Design team was unable to test the Robosub this semester due to the current global pandemic.

**3. External Interface Requirements**

**3.1 User Interfaces**

The Controls subsystem does not require any user interface due to the the fact that the Robosub will be running autonomously.

**3.2 Hardware Interfaces**

The Controls subsystem will require an Arduino microcontroller to communicate with the eight thrusters on the Robosub. It will also be interacting with another subsystem to collect data from the IMU sensors.

**3.3 Software Interfaces**

We will be using Ubuntu version 18.04, as well as ROS version Melodic. The Controls subsystem will also use the PID libraries in the Arduino IDE.

**3.4 Communications Interfaces**

The Arduino microcontroller will communicate with the thrusters and the IMU sensors, and is supported with C, which will be implemented later using the Arduino IDE.

**4. Requirements Specification**

**4.1 Functional Requirements**

This section lists the functional requirements of the Controls subsystem. This section will also include the complete set of functional requirements, along with explanations for cases in which the statement of the requirement was deemed insufficient or requires additional clarification. All requirements will relate to the design modules described in the Design Document.

|  |  |
| --- | --- |
| **Requirement No.** | **Requirement Description** |
| 4.1.1 | The Arduino code shall output proper speed to its thrusters so it can balance the AUV. |
| 4.1.2 | The Arduino code shall subscribe from ROS topics of pitch and roll angles to determine if the Arduino can make it balance with angle 0 degree. |
| 4.1.3 | The Arduino code shall output proper speed to its motors to move forward or backward as it receives input from ROS navigation system. |

**4.2 External Interface Requirements**

|  |  |
| --- | --- |
| **Requirement No.** | **Requirement Description** |
| 4.2.1 | The system shall handle data using ROS’ publishers and subscribers. |
| 4.2.2 | The system shall interface the AUV’s subsystems together to operate the AUV. |
| 4.2.3 | The system shall output various movement commands to the Controls subsystem for the AUV’s mobility. |

**4.3 Logical Database Requirements**

The Controls software does not have any database requirements.

**4.4 Design Constraints**

We currently do not have any additional design constraints that require technical descriptions.

**5. Other Nonfunctional Requirements**

**5.1 Performance Requirements**

* The Robosub has one terminal.
* The Robosub supports two simultaneous users at once.
* The amount of information that is handled by the Controls subsystem is dependent on the data that is received from the Navigation and Mission Planning subsystems of the Robosub.

**5.2 Safety Requirements**

One possible loss or damage that could result from the Robosub is that it could collide and harm another diver within the AUV’s reach. It could also potentially damage itself if it spins too fast or gets out of balance, causing significant damages to the equipment.

**5.3 Security Requirements**

There are no security requirements needed for the Controls subsystem.

**5.4 Software Quality Attributes**

There are no software quality attributes for the Controls subsystem at the moment.

**5.5 Business Rules**

Any software developer with basic knowledge of the PID controller and the Arduino IDE, along with the knowledge of the C language can perform the functions of the product.

**6. Legal and Ethical Considerations**

The Controls subsystem faces legal and ethical issues with the project when it comes to releasing our developed code to the public, since it’s used in a competition to compete with other schools.

**Appendix A: Glossary**

See Section 1.4 for abbreviations.

**Appendix B: Analysis Models**

See Section 2.2 and Section 2.7 for the Controls flowchart and block diagrams.

**Appendix C: To Be Determined List**