Technical Design Report

Trident Delft Roboboat Competition Entry

Stefan Bonhof
Mechanical Engineering

Remi Flinterman
Computer Science and Engineering

Bob van Rooij
Mechanical Engineering

Izaak Cornelis
Electrical Engineering

Tim Puglia
Industrial Design Engineering

Albert Smit
Computer Science and Engineering

***Abstract*—This document contains the overall competition strategy of Trident Delft. It will cover the design process behind the Trident itself and the way in which the team plans on winning the competition.**

Fig. 1: Side view of the Trident

I. Introduction

Trident Delft started off as a project-group for the Robotics minor at the Delft University of Technology. This is the first time that a team from the TU Delft will participate in the Roboboat Competition. Except for some knowledge on manufacturing a fiberglass hull, the team started with zero experience. Overall it has been a very exciting process and this report will provide an overview of how the team handled the challenges.

II. Competition Strategy

In order to win, a number of choices had to be made regarding which challenges to compete in, and to what degree they should be finished. This section will provide insight into how the team plans to do this. Initially, the subsystems used in the challenges will be presented, followed by a short explanation of how each challenge will be participated in.

A. System complexity and design approach

The design of the vessel is relatively simple. During the production phase, a majority of the time went into the hull, as each layer of fiberglass needed time to dry. All other systems are however mounted fixed, so for example, no complex parts for moving motors were needed. By keeping everything simple and by designing the system with modularity in mind the team could, in case of an emergency, replace parts of the system in a simple and efficient manner.

Besides being modular the vessel is also designed to be very light. The first iteration was 22 kilograms (48.5 pounds) but the current vessel only weighs 12.5 kilograms (27.5 pounds). By keeping the weight low the team hopes to score well on the thrust-to-weight test. The size of the vessel has been down-scaled in such a way that all of the electronics fit perfectly and so that no space is wasted. This will result in easier navigation through even the smallest gates.

B. Object detection and identification

The system as a whole needs to know where the obstacles in its vicinity are in order to be able to avoid making contact with them. Thus, an obstacle detection system is required. The vessel will use this data to generate a map so that a path can be planned. The Trident shall achieve this using its lidar system. To make sure the vessel knows what kind of object it is dealing with a camera will also be present in order to distinguish buoys and objects based on color and shape.

C. Positioning

The positioning system aims to combine data from several sensors in order to derive the position of the vessel relative to obstacles in its surroundings. This is called Simultaneous Localization And Mapping (SLAM). Seeing as the GPS sensor has quite a large deviation when compared to the actual size of the vessel [2], an IMU will also be placed on the vessel to make sure it can determine its position more accurately. Once the vessel knows where it is and where it should go the computer will provide the thrusters with the appropriate power to go from A to B. The vessel steers using differential thrust. The moment generated between the two stationary motors will provide the vessel with a moment that it can use to make turns.

Challenge 1: Autonomous navigation

From the starting GPS coordinates, the vessel will look for the first gate and once found it shall move to a point at a fixed distance from said gate. Once the vessel is in position and ready to do the challenge, the vessel will start accelerating and pass through the first gate. The vessel will maintain a heading perpendicular to the start gate buoys. Once past the start gate buoys, the vessel will look for, locate and classify the end gate buoys. After moving through this gate the vessel will proceed to the next challenge.

Challenge 2: Speed challenge

This challenge will be completed in a manner very similar to the Autonomous Navigation challenge. Instead of looking for a second gate the vessel will look for a blue A2 marker buoy, circle it counter-clockwise and return through the gate through which it entered.

Challenge 3: Find the path

In order to plan a path through the field of buoys, the vessel will start looking for the white can-buoy in the center of the field using its camera. Once located the computer will plan a path to a point near this buoy. Once there the vessel will circle the buoy and look for the nearest way out.

Challenge 4: Automated docking

Due to an unexpected delay in funding the team will not be able to attempt this challenge to its full extent. If last-minute funding becomes available the team will attempt to purchase a drone which will take off and land during this challenge. However, time does not allow for the programming of both a drone and a hydrophone in the last days before the team departs to the USA. The vessel will attempt to enter a random bay and perform a quick drone launch and retrieval in order to gain points even though no hydrophone is present in the vessel.

Challenge 5: Follow the leader

The vessel will distinguish the flag from other objects using its color tracking software. The color tracker will look for the edges of the flag and use this to determine its location. Once it is located the computer will make sure the vessel stays behind the flag for a full revolution of the carousel.

III. The Trident

The Trident does not only consist of a number of subsystems. Several practical choices regarding its shape and the placement of the subsystems had to be made as well, these will be discussed in this section.

A. Hull selection

Generally, there are two types of hulls: planing hulls and displacement hulls. Planing hulls achieve high speed by lifting out of the water to reduce the area of friction. This means that the drag on the boat is drastically reduced, facilitating a higher top speed. However, planing hulls have two significant drawbacks: they only work efficiently at high speeds and they are relatively unstable. Displacement hulls do not rise out of the water, regardless of speed, so they are more stable and do not require a high speed in order to operate. As stability is of great importance this is the type of hull the team will use.

Regarding the shape of the hull, three options are considered: a mono-hull, a catamaran, and a trimaran. Mono-hull boats are the most traditional. They are easy to produce, however, they can get unstable easily. The catamaran and trimaran designs are more stable than the mono-hulls and are used by most other competing teams as well. These multi-hulls do lose some maneuverability when compared to the mono-hulls. However, seeing as stability is more important to the team than maneuverability, these multi-hulls are actually preferable. While performance-wise the catamaran and trimaran are very similar, the center hull of a trimaran allows for more storage room for electronics and facilitates easier weight distribution.

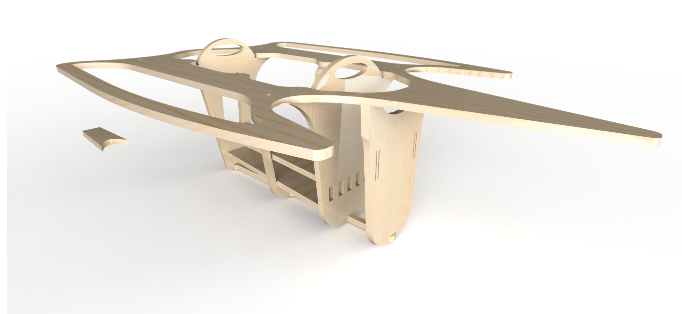

Fig. 2: Support structure of the Trident

B. Hull production

In order to create a model with the smooth shape of the Trident, the hull was made out of glass fibre. The reason glass fibre was selected over other fibers, such as carbon, was mainly the price. By applying sheets of glass fibre and then coating those with an epoxy resin, a strong hull was constructed. Glass fibre sheets are easy to handle and the process turned out to be very forgiving. Small mistakes were easily corrected by simply applying more fibre and epoxy resin. In order to even out the last imperfections a coat of putty was applied before the hull was painted. A lightweight wooden frame has also been installed to fortify the hull and keep it from bending or warping while in the water. This frame can be seen in figure 2.

C. Component placement

In figure 3 the position of the lidar and camera can be seen. The camera is positioned high enough for it to be able to register the flag in the follow-the-leader challenge. Because the camera used by the team is very small, they were able to place it inside of the mount for the lidar, reducing its exposure to water. The lidar has been placed upside down, so the plane in which it scans is as close to the water as possible so it can detect even the smallest A0-type buoys.

In order to keep the computer systems and non-vision sensors compact and easily accessible these where mounted on a frame that from now on shall be referred to as ”the suitcase”. It can be seen in image 4. The lidar and radio are connected

Fig. 3: Lidar and Camera placement

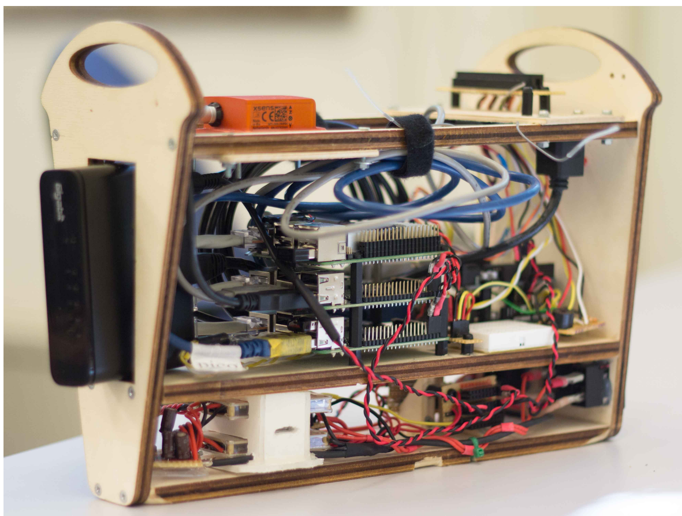

Fig. 4: The Trident's computer "suitcase"

with their own cable and can be easily disconnected. All other sensors on the outside of the hull are connected to the suitcase with a 40-pin ribbon cable so it can be taken out simply by easily.

D. Towing and lifting harness

As stated by the competition rules each vessel should possess a frame that can be used for thrust-measurements, towing and lifting by a crane. Nuts with rubber washers will be fastened to the bolts that are used to connect the topside of the Trident with its hull. Carabiners can be looped through these rings for the appropriate task at hand.

IV. Computer and Software Structure

The Trident is running ROS Kinetic on Ubuntu 16.04. Python was used as the main programming language for most of the software, the Arduino boards and the node that covers buoy mapping, which will be discussed in sectionIV-A6, were coded in C++.

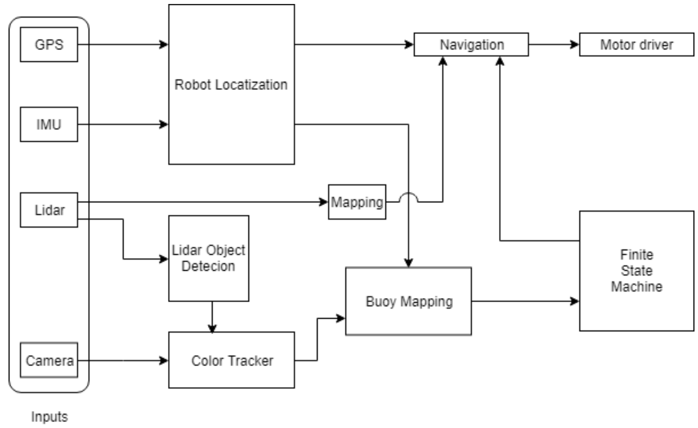

Fig. 5: Software architecture of the Trident

A. Software architecture

In order for the Trident to operate the team has constructed the software architecture seen in figure 5. A quick overview of what is being processed in each "node" will be provided in this section.

1) Inputs: The input blocks each manage a sensor. Each converts the data from it's corresponding sensor into ROS messages that can be used further down the line.

2) Robot Localization: This is where the robot determines it's own location. By combining the GPS and IMU data using a Kalman filter [1] the overall accuracy of the boat's location is increased so it knows where it is. It allows location data to be transformed between a number of frames, local and global.

3) Lidar Object Detection: The lidar object detection node checks the data it receives from the lidar input and compares the shapes it registers with the shape of a circle. If it is similar enough it will be registered as a buoy and published as such. Each buoy will also be given a type based on it's dimensions.

4) Mapping node: This node works closely with the navigation node. It does not map buoys as colors or types, it only provides a map with obstacles and areas around said obstacles that the vessel should not move in to.

5) Color tracker: This node uses OpenCV to recognize the colors of the objects the vessel can see. It links the colors it sees with the objects the lidar object detection node detected and sends this information to the Buoy-mapping node.

6) Buoy Mapping: This node combines the transformed locations from the Robot localization node with the complete list of buoys received from the Color tracker. This map provides data for the Finite State Machine (FSM) for it's decision-making.

7) Finite State Machine: The actual decision-making of the vessel happens in this machine. It contains a high-level FSM which consists of the challenges the vessel has to perform and each challenge itself has also been converted into a smaller FSM. The network was created using Flexbe, a library that allows for the easy integration of states into larger behaviors. It also provides a very useful means of saving data on a global level accessible to all states in the FSM. The node sends out goals whenever it has found one to the navigation node.

8) Navigation: In this node the ROS navigation package takes the goals from the FSM and the data from the mapping node (not to be confused with the buoy mapping node) and plans a path. In doing so it avoids any objects and it outputs a velocity command.

9) Motor driver: In order for the thrusters of the Trident to give an output the velocity commands from the Navigation node are converted to motor-speed commands.

B. Hardware

As can be seen in Appendix A, the Trident houses quite a number of computers. This section will cover what is being processed on the Trident and where.

As of now, the UP board is being used to process the color tracker. Initially, the team attempted this on one of the Raspberry Pi's, however, these did not have enough processing power.

The Pi's are used for the main sensor inputs. The FSM that controls the vessels decision-making also runs here.

Finally, there are the Arduino boards. The first one is used to monitor the battery level and read out the water sensors in the bottom of the hull. The second board is used for the radio module. The third and final board is used as a motor driver. This board is also connected to the emergency stop.

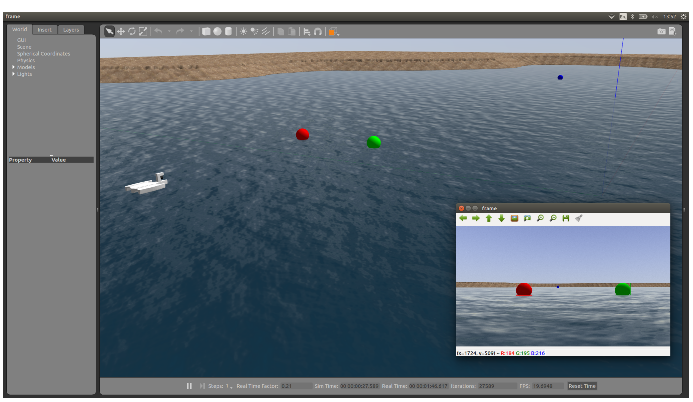

Fig. 6: Gazebo environment

V. Experimental Results

In order to prepare for the competition, the team tested the vessel both in real life and in a virtual environment.

A. Digital testing

In order to test the Tridents autonomous capabilities, the team constructed a virtual environment which can be seen in figure 6. A Gazebo world was created with working dynamics, all the sensors of the Trident and the actual challenges from the competition. By using the actual ROS software running on the real vessel in the simulation all challenges were performed digitally before actually placing the real vessel in water.

B. Real life testing

The vessel has been in actual water for a number of times. Most initial tests were on stability, maneuverability, and weight distribution. The camera and the lidar were tested thoroughly with the Color Tracker, Lidar Object Detection, and Buoy Mapping nodes in order to learn more about how the system responds to different shapes, ranges, and light intensities. At the time of publishing the system as a whole has not been in the water in its fully autonomous state yet, however, this will be attempted multiple times before the competition starts.

Acknowledgment

The Trident would have never been possible without help from outside. First of all the team would like to thank the Damen representatives that helped during both the minor and the period leading up to the competition. Special thanks to Siebe, Carola, and Sanne, without them the team would have never been able to participate. The team would also like to give a special thanks to Martin Klomp of Robovalley. He coached the team from beginning to end and provided a lot of useful insights.

Finally, the team is very grateful to their sponsors and partners. Seascape, Xsens, StuD, Defensity College and the Department of Maritime Transport Technology of the TU Delft provided the team with the funds and parts necessary to actually build the vessel, without them there would be no Trident.

References

[1] R. Faragher,Understanding the basis of the kalman filter via a simple and intuitive derivation,(2012)

[2] Adafruit,Adafruit ultimate gps breakout - 66 channel w/10 hz updates - version 3,(2018),[Online;last checked 04, June 2018]

Appendix A: Component Specifications

Component	Vendor	Model/Type	Specs	Cost (if new in euros)
ASV Hull form/platform	Trident Delft	Trident	Trimaran	-
Propulsion	Seascape	Bluerobotics T200	https://bit.ly/2IE0w1N	143.00
Power System	Hobbyking	Multistar High Capacity 4S 16000mAh Lipo	https://bit.ly/2KJL8By	86.00
Motor controls	Seascape	Bluerobotics ESC	https://bit.ly/2IDHEQm	25.00
Computers	Pre-owned	Raspberry Pi 3b (2x)	https://bit.ly/1WTq1N4	40.00
	Pre-owned	Raspberry Pi 2b	https://bit.ly/1b75SRj	30
	Pre-owned	Arduino Uno (3x)	https://bit.ly/2rk4y6m	24
	Pre-owned	UP board x1	https://bit.ly/2KQY3Sm	90
Inertial Measurement Unit (IMU)	Xsens	MTi-300-AHRS-2A8G4-DK	https://bit.ly/2s18WJy	2900.00
Global Positioning System (GPS)	Antratek	Ultimate GPS breakout v3	https://bit.ly/29taSjM	36.00
Camera(s)	Pre-owned	Pi camera module V1	https://bit.ly/2x5h3us	25.00
Lidar	Manu-systems	Hokuyo UTM-30LX-EW	https://bit.ly/2LjDnnf	5000.00
Radio	Mouser	Semtech SX1272MB2DAS (2x)	https://bit.ly/2kuRhGC	21.00
Aerial vehicle platform		Parrot Bebop 2	https://bit.ly/2AzxCvO	500
Team Size (number of people)	6			
Expertise ratio (hardware vs. software)	50/50			
Testing time: simulation	50 hours			
Testing time: in-water				
Programming Language(s)	Python C++			