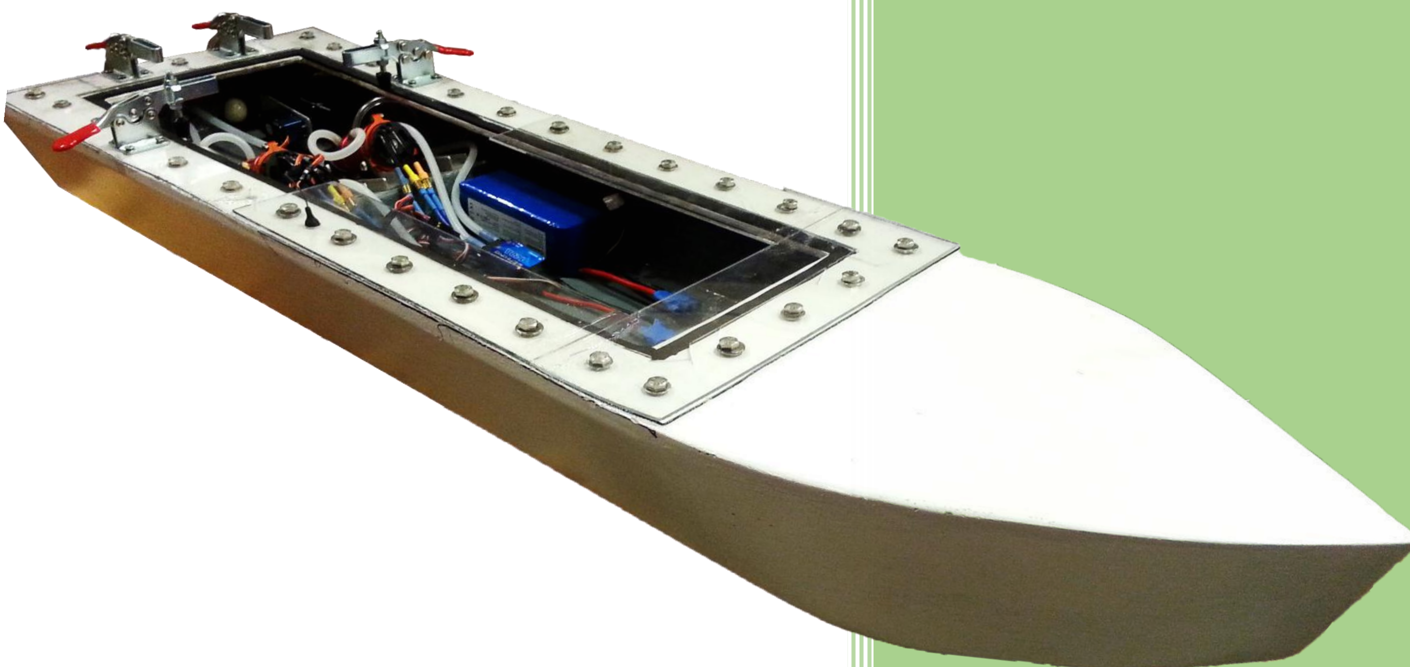
Autonomous Surface Vehicle

Andrew D'Amore, Kristin Therrien,
William Lerner, Tomas Conway,
Michael Bogochow, Cody Noel,
Jeffery Masucci



Department of Mechanical Engineering
University of New Hampshire
24 April 2014

ABSTRACT

The main objective of the Autonomous Surface Vehicle (ASV) project is a proof of concept of autonomy for a surface vehicle with logic that can be implemented on various surface vehicles. To accomplish this a small scale prototype, about the size of a large RC boat, is developed in order to implement the autonomy. The autonomy is accomplished by using the MOOS-IvP program developed by Harvard and … through a BeagleBone Black microprocessor. Autonomy also requires a feedback system to relay relevant information back to the MOOS-IvP program such that it can make decisions. The feedback sensor suite on the ASV includes: encoders for RPM feedback, a 9 degree of freedom IMU for heading feedback, ultrasonic sensors for obstacle feedback, a PlayStation®Eye camera for tracking feedback, and a GPS for position feedback. 3 specific tasks are to be completed by the ASV including: traveling to a specified waypoint from any random location, track and trail, and obstacle avoidance.

Table of Contents

LIST OF FIGURES AND TABLES

ACKNOWLEDGEMENTS

The group would like to acknowledge Sheldon Parent, Paul Lavoie, Scott Cambell, Mike Conway, and James Abare for allowing us to use their facilities and helping us with various construction tasks. Additional thanks to the UNH NASA MMS QuadSatC team for developing the visual recognition program. We'd also like to thank Tracey Harvey and Jenn Bedsole for their valuable input on our project.

INTRODUCTION

Autonomous Surface Vehicles (ASV) are extremely useful in both civilian and military applications. ASVs can be used for tasks where using a human operator is to inefficient compared to robot such as ocean mapping, displayed in Figure 1. ASVs are also useful in situations where having a human operator is too dangerous such as: surveillance and reconnaissance of hostile areas, search and rescue missions, and defense and early warning against submerged hostile vessels. Due to the lack of autonomous technology commercially available, research is focused on a proof of concept to autonomy.

To accomplish these three autonomous tasks are to be performed by the ASV. The first task, getting to a specified waypoint from a random location is the easiest to perform if GPS is available to

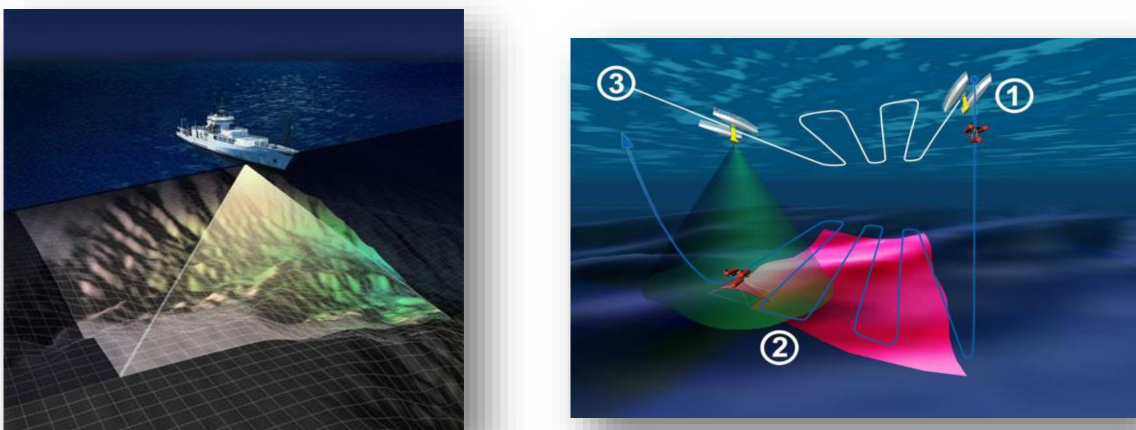


FIGURE 1: (LEFT) ASV PERFORMING OCEAN MAPPING. (RIGHT) ASV TRAILING A REMOTELY OPERATED VECHICLE (ROV) PERFORMING OCEAN MAPPING. [3]

use. However this task is the most crucial as it requires the ASV to know its current location as well as the location of its objective, which is necessary for almost any more complicated tasks. The second task, track and trail, can be accomplished without GPS (location feedback) as it only requires the ASV to know its location with respect to what it is tracking, allowing for a wider variety of use due to not being constructed by GPS. The third task: obstacle avoidance is arguably the second most important task for the ASV as no matter what task the ASV is trying to accomplish it needs to be able to react to any unforeseen obstacles.

To accomplish Task 1 the ASV includes encoders for RPM feedback, a 9 degree of freedom IMU for heading feedback, and a GPS for position feedback. The sensors used for Tasks 2 and 3 are a PlayStation®Eye camera for tracking feedback, and ultrasonic sensors for obstacle feedback respectively. All of these tasks require a program to constantly be taking in the current situations data, processing it, and make a decision on what it would like the ASV to do, which is accomplished by MOOS-IvP. The Experimental Detail section of this report will further explain how the prototype is constructed, how the sensors integrate into the system, and how MOOS-IvP functions and makes decisions. The Future Work section will discuss the immediate and long-term future of the project. The Results and Discussions section will go into detail about the conclusions.

EXPERIEMENTAL DETAILS

Hull

The hull of the autonomous surface vehicle is comprised of three main components; the bottom, the top, and the top latch. The bottom of the hull is a V-shape and is chosen due to its prevalence in the maritime community. The particular V-shape of the hull is a scaled up version of a typical remote control (RC) boat as most RC boats do not leave the necessary room to install and update the electronics needed to control the autonomy of the boat. The top section of the hull has a large open section to allow for the necessary modifications. The latch of the hull is implemented to create a water tight seal over the large opening in the top section of the hull to keep the electronics inside dry and functioning while still allowing the electronics to be accessed and modified.

Making the hull out of fiber glass is the best option because it minimizes the weight while still maintaining a high level of strength. The bottom section of the hull is constructed using a three step process which includes two molds required to construct the final hull. The first mold is constructed using sheets of welded steel to create a rigid framework for the second mold. For this second mold putty is applied to the outside of the model and then it is sanded down in order to create a smoother surface on which to create the second mold. The third step is to put fiberglass on the outside of the first mold to create the second mold. In order to remove the fiberglass mold from the bottom of the steel mold an air hose tap is added to the inside of the steel mold to apply pressure and remove the fiberglass mold. From the first fiberglass mold, the final fiberglass hull is constructed. More fiberglass is applied to the inside of the first fiberglass mold in order to create a smooth surface on the outside of the final hull. The final hull is then removed from the fiberglass mold utilizing the same pressure tap. The steel (1) and fiberglass (2) molds and the final bottom portion of the hull (3) are shown in Figure 2 below. The steps are in order in the photo starting from the left and finishing on the right. Additional pictures of the process can be found in Appendix A.

FIGURE 2: HULL CONSTRUCTION PROCESS. FROM LEFT TO RIGHT, STEEL MOLD, FIBERGLASS MOLD, TO FINAL FIBERGLASS HULL.

Once the bottom of the hull is sanded smooth, the top section of the hull is ready to be created. The top section is chosen to be a thin strip of fiber glass with a large rectangular opening in order to leave room for the installation of a removable latch. The top section of the hull only requires one mold which is then used to create the fiberglass top. The top section of the boat is then attached to the bottom and water sealed with another later of fiberglass. Figure 3 below shows the top and bottom sections of the hull before (left) and after (right) they are fiber glassed together.

FIGURE 3: (LEFT) UNSEALED TOP AND BOTTOM HULL COMPONENTS. (RIGHT) SEALED TOP AND BOTTOM.

After the two pieces of the hull are combined, a motor mount is installed to provide an angle and more stability of the motors. Along with the motor mount, two rigid rods are installed on the bottom of the hull at an angle corresponding to the motor mount using fiber glass steel rods. These rigid rods have a hollow center in order to house the flex shafts which connect to the propellers and are necessary to keep the flex shafts in a stable position as well as maintain a marine grease layer in-between the flex shafts and rigid rods to prevent frictional losses and water seal the inside of the rods. Along with the motor mount and rods, two shelves are fiber glassed into the hull. The shelf in the front provides adequate room for the electronics and the back shelf provides housing for the servo motors used to control the rudders. Along with housing the electronics, the shelves in the boat also provide stability and strength to the structure as a whole.

Motors

A dual motor system is chosen to supply sufficient power and thrust due to the size and weight of the ASV. Two leopard LBP4074/4Y 4-pole in-runner motors are mounted into the hull of the ASV. The LBP4074/4Y motors have a maximum power of 2600W, maximum amperage rating of 65A, and provide 1050kV (1 kV = RPM/Volt). Due to the prototype only needing proof of concept for autonomy, it is decided that the ASV will operate at low speeds for safety and ease of programming. Additionally the ASV has a high weight and size, therefor requiring a relatively high torque from the motors to move the system. For the same power rating, a lower kV rating will output a greater torque. The "4Y' model is chosen over the "2Y", "2.5Y", and "3Y" models to supply the greatest possible torque for the ASV while sacrificing maximum RPMs as a high maximum speed is not desired.

The angle (θ) from the prop to the perpendicular axis of the water surface determines the forward thrust of the correlating motor, shown in Figure 4. The greater θ the greater the forward thrust of the ASV. The angle at which the motors are mounted is chosen to provide enough clearance for the rudders while providing the maximum forward thrust to the ASV. Seen in Figure 4 the flex rod (the rod connecting the

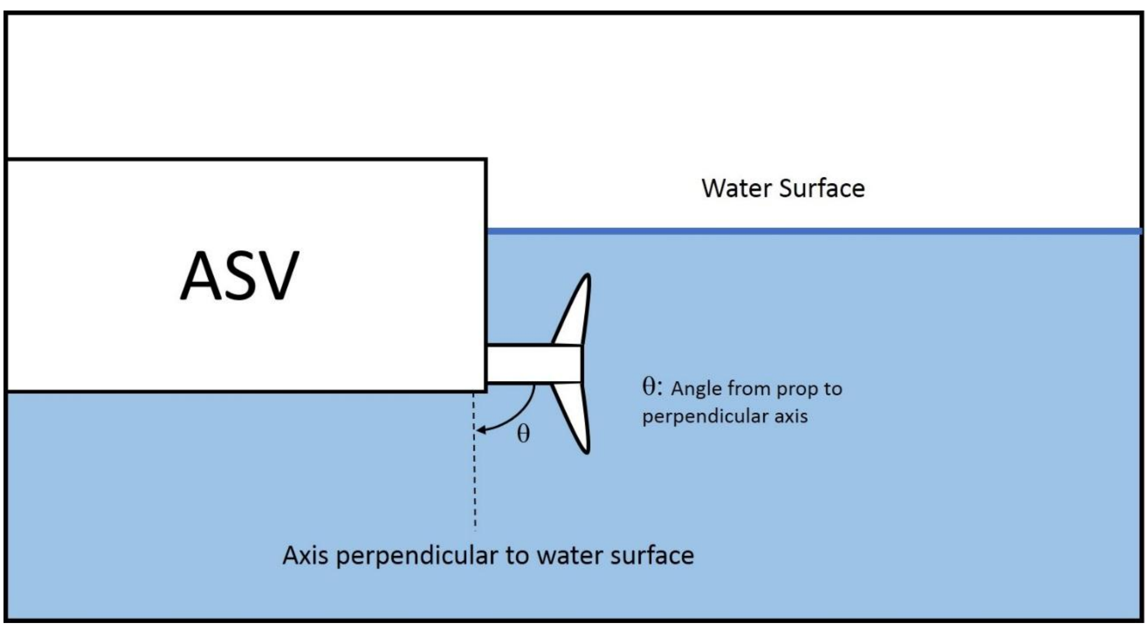


FIGURE 4: DIAGRAM OF ANGLE OF PROP TO PERPENDICULAR AXIS

output shaft of the motor to the props) exits the ASV below the hull through the rigid steel rods installed into the hull.

FIGURE 5: MOTOR AND RUDDER SETUP OF ASV.

Each motor is connected to a Hobby Wing SeaKing 120 ESC. The maximum amperage rating of the ESC is 120A to safely insure that the batteries will not damage other components. The purpose an ESC is to vary 3-phase electric motor's speed, typically on a radio controlled vehicle model. The receiver on the ASV will pass a value from the throttle of the controller to the ESC, the ESC will then use pulse width modulation (PWM) to convert the throttle value to a rotational velocity. As the value from the receiver increases, the frequency of the signal (PWM of the signal) sent to the motors from the ESC will increase, resulting in a faster rotational velocity.

A cooling system is implemented to keep the motors at a low constant temperature in order to insure the internal electrical components do not over heat. The motor mount contains a cooling port to keep the temperature of the motors output shaft low. After exiting the cooling mount, the water enters a second cooling jacket that surrounds the motor to keep the majority of the motor cool. The water then passes through a heat sink built into the electronic speed control (ESC) before exiting the ASV at the rear of the hull.

Rudders

A dual rudder system is implemented on the ASV to provide an increased control of the vessel as seen in figure 6. Due to the large length of the ASV (48”) large scale rudders are chosen. The mission objectives require the ASV to avoid obstacles, track and trail targets, and to proceed to way points; all of which are highly dependent on vehicle control and precision, both of which are increased with increasing rudder size. Each rudder unit is controlled by a single servo motor within the ASV seen in Figure 6, which attaches to a pushrod shaft that passes through a water-tight rubber fitting in the hull to connect to the rudder and prevent water from entering and damaging the internal components of the ASV.

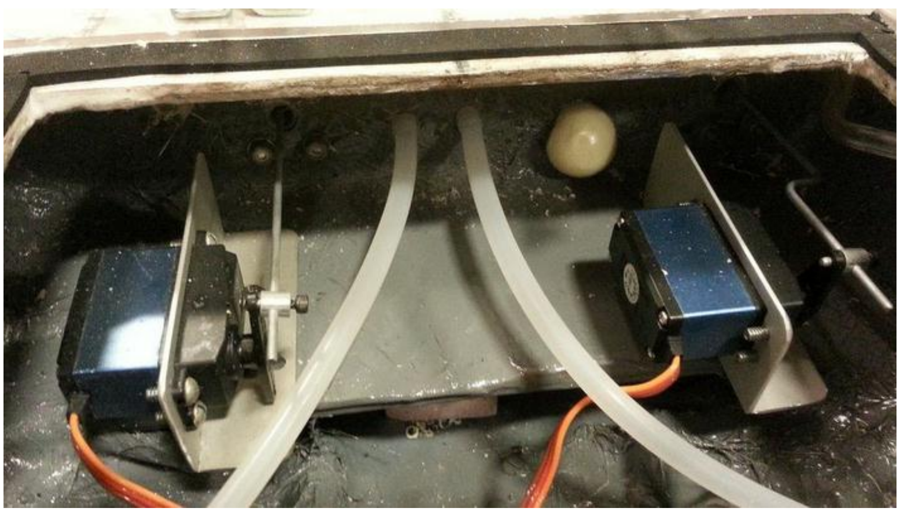


FIGURE 6: RUDDER SERVO ASSEMBLY.

The rudders serve an additional purpose aside from vehicle control. As the ASV moves forwards, pressure taps located on lower section of the rudder fins (viewed clearly in Figure 7 have water pushed through the rudder and enter the water cooling system though tubes connected on the top of each rudder unit.

Figure 7: Rudder assembly

Latch

The latch of the ASV is regarded as highly important to assure that no water enters the hull. Any water within the hull of the ASV is seen as a risk and can critically damage the internal electrical components which control the direction and movements of the vessel.

Figure 8: ASV latch

After researching latch possibilities commonly used in model RC boats, the most practical method of sealing the ASV is to rest the latch on top of a water-resistant neoprene tape, and apply pressure along the edges to provide a complete seal.

Figure 8 displays the overall latch design implemented on the ASV. Plexi-glass is bolted around the opening in the top of the hull to create a casing for the latch to minimize any possible gaps from imperfections on the top of the hull. Silicon is then used to seal the edges of the casing to eliminate the possibility of any water under the plexi-glass. Additional pleix-glass is bolted to the front end of the hole and the latch slides under this overlapping plexi-glass to provide constant pressure to the front of the latch. Four HH-201B Horizontal Handle Toggle Clamps are used to apply pressure to the rear portion of the casing. Each clamp has a holding capacity of 200 pounds.

MOOS-IvP

The autonomous functions for the vehicle are carried out using an open source software suite called MOOS-IvP, composed of two components, collaborated by Oxford and MIT. The suite as a whole is specifically designed for autonomous control of marine vehicles and is composed of two main components being MOOS (Mission Oriented Operating Suite) and the IvP Helm (Interval Programming Helm). MOOS allows multiple special programs, known as MOOSApps, each contributing to a part of the autonomy, to communicate with each other through a publish/subscribe database known as the MOOSDB. The most important of these programs is the IvP Helm which handles the actual autonomous decision making based on competing autonomous behaviors.

MOOS-IvP can be downloaded via a subversion repository and information regarding the software suite can be found at moos-ivp.org.

MOOSApps

MOOSApps are special programs which are connected to a single MOOSDB; a set of MOOSApps connected to a single MOOSDB is known as a MOOS Community, as shown in Figure 9. MOOSApps never communicate directly with each other, but instead post to and read from the MOOSDB. This allows for there to be no dependence of one MOOSApp to any other MOOSApps and so any combination of MOOSApps can be used for any given mission.

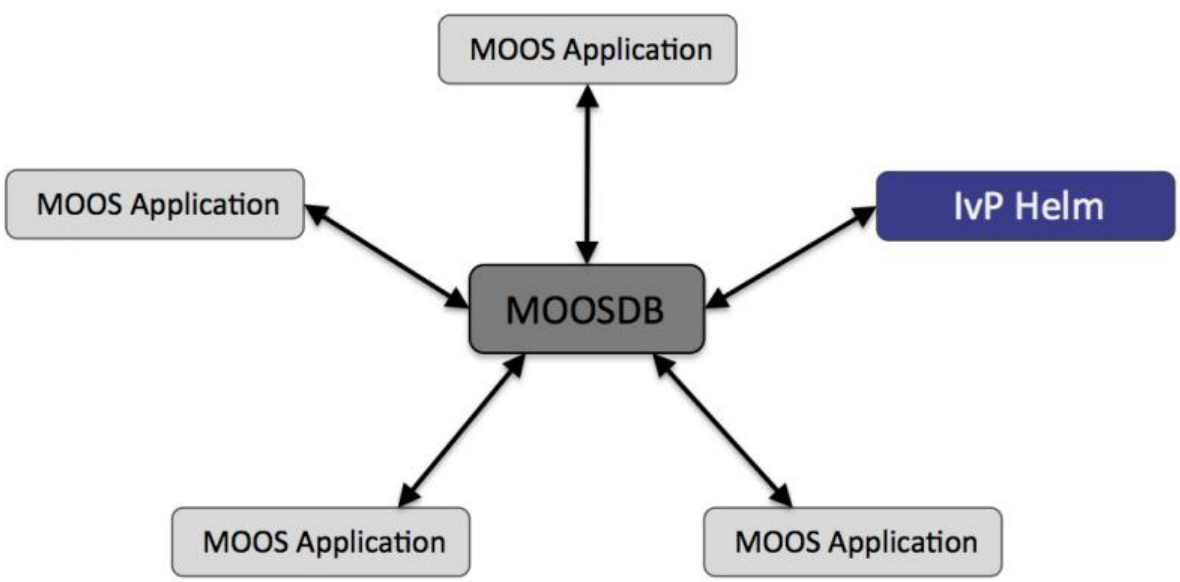


FIGURE 9: EXAMPLE OF A MOOS COMMUNITY CONTAINING THE IVP HELM. [2]

A MOOSApp will subscribe to a set of fields in the MOOSDB on launch. Whenever there is a change to one of a MOOSApp's subscribed fields, no matter what the source of the change was, the MOOSApp will be updated with the new value of the field. A MOOSApp also has the ability to post new data to any field in the MOOSDB.

A MOOSApp can perform any function. MOOS-IvP comes with a variety of MOOSApps. One such MOOSApp is the pNodeReporter which collects navigation information from the MOOSDB and wraps it into a summarizing report known as a NODE_REPORT. These NODE_REPORT messages can be consumed by other MOOSApps and are especially useful when sending data between MOOS communities. Another MOOSApp packaged with MOOS-IvP is the pMarineViewer. The pMarineViewer MOOSApp is a graphical user interface which displays a variety of data being used internally by MOOS-IvP as well as a graphical interpretation of the area of operations including the locations of the vehicles and their waypoints and paths. The pMarineViewer MOOSApp can be seen in Figure 10.

MOOSApp Architecture

All MOOSApps have the structure showed in Figure 10. The OnStartUp method is called once after the process starts and is used for registering for entries in the MOOSDB. After this method returns, the

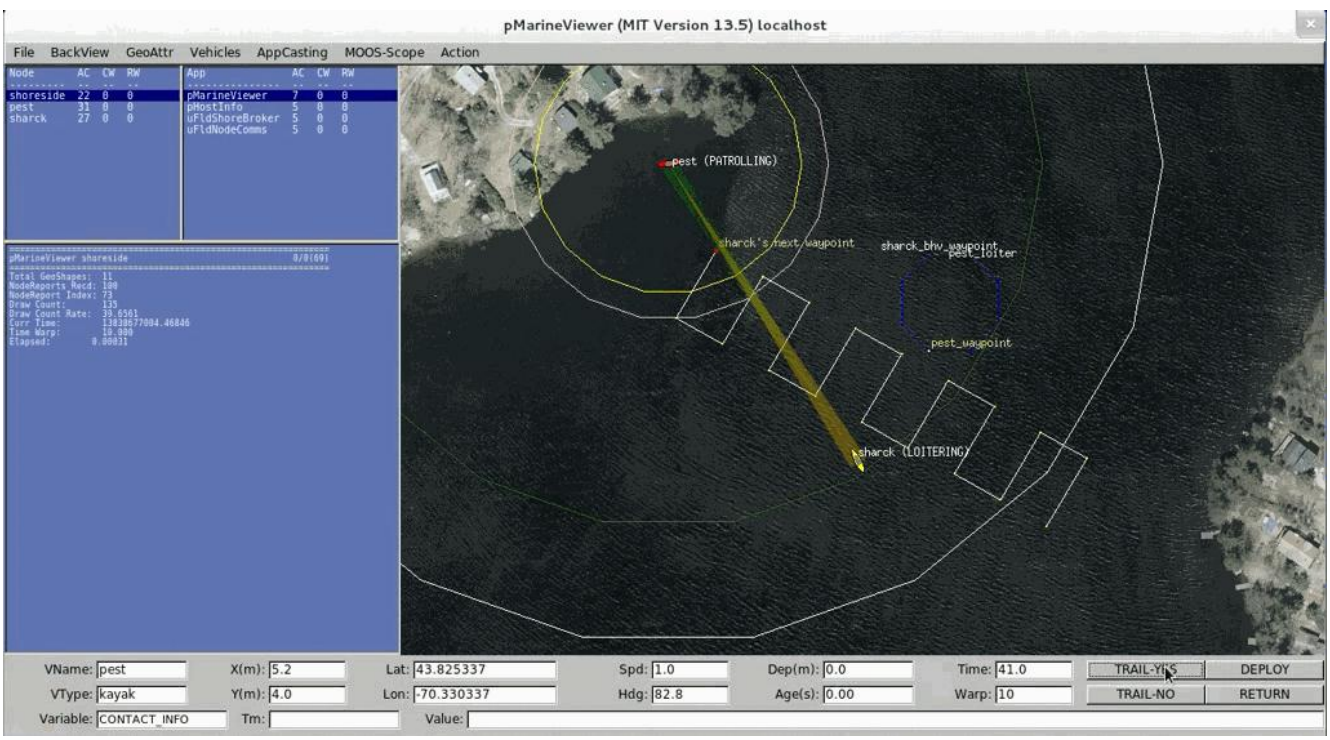


FIGURE 10: THE BUILT-IN MOOS-IVP GRAPHICAL MISSION VIEWER (PMARINEVIEWER) RUNNING A SIMULATION OF A TRACK AND TRAIL MISSION.

MOOSApp goes into an infinite loop of calling the OnNewMail and then the Iterate method. The OnNewMail method is used to read from the MOOSDB. The Iterate method is where the MOOSApp executes its main algorithm for achieving the purpose of the MOOSApp.

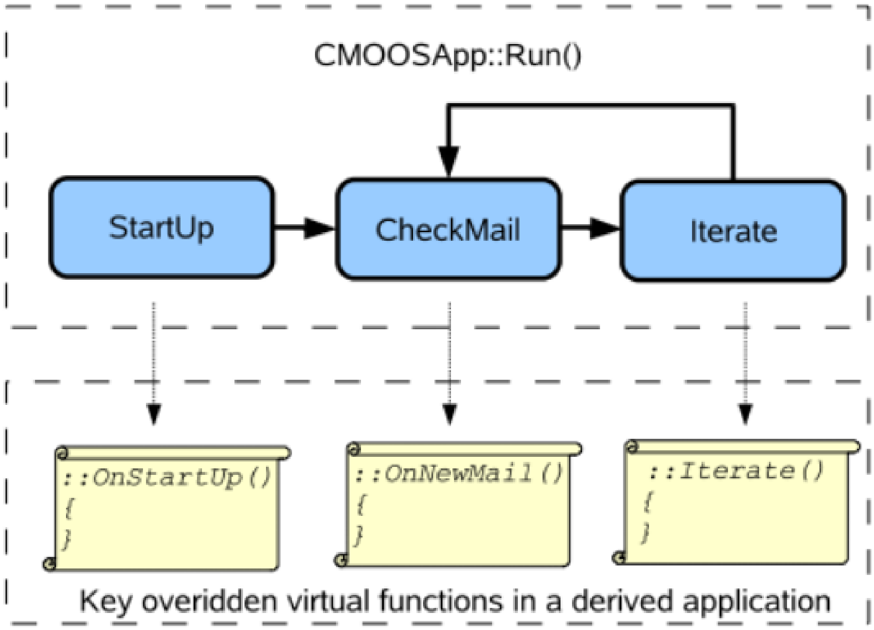


FIGURE 11: THE MAIN MOOSAPP PROGRAM FLOW AND THE CORRESPONDING METHODS WHICH PERFORM THE PROGRAM EXECUTION AT EACH STATE. [2]

All MOOSApps extend the CMOOSApp class provided with MOOS, as show in Figure 11. The CMOOSApp handles all of the structure of a MOOSApp including setting up and executing the mail-iterate loop. The only methods that a MOOSApp needs to overwrite are the Iterate, OnNewMail, and OnStartUp to perform the functions described above. There is also an OnConnectToServer method which can be overridden. This method is called whenever a connection to a MOOSDB is made and can be used to register for entries in the MOOSDB. It can be useful to do this in cases where connection to the MOOSDB is lost and then later reestablished.

The IvP Helm

The IvP Helm is a single MOOSApp which is responsible for making autonomous decisions. It utilizes a behavior-based architecture in which multiple prioritized "behaviors" are competing for control of the vehicle at any given point. Autonomous behaviors are certain actions which the vehicle should take during a mission. Behaviors can be single actions like navigating to a certain location or they can be continuous like avoiding obstacles. When multiple behaviors have input to control the vehicle, the IvP Helm decides on a single output to the vehicle controls based on these behaviors. The output may solely be one of the behavior's desired output or it may be a compromise between several different behaviors.

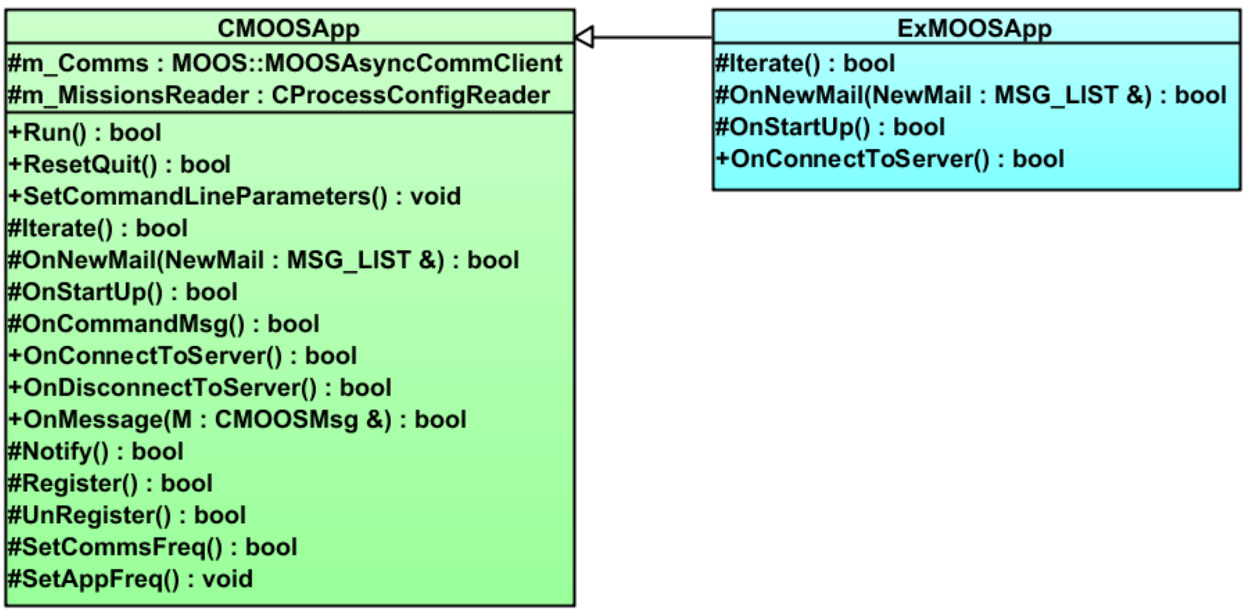


FIGURE 12: CMOOSAPP INHERITANCE CLASS DIAGRAM. [2]

IvP Behaviors Example

As an example of behavior reconciliation by the IvP Helm, suppose there are two active behaviors, a waypoint behavior which says that the vehicle should travel to a set location, and an avoid obstacles behavior which says that the vehicle should avoid contact with any obstacles that it detects. The waypoint behavior will always attempt to control the vehicle in a way that will make it reach the waypoint while traveling the shortest distance to get there as possible. However, the avoid obstacles behavior will cause the vehicle's path to be altered if any obstacles are obstructing the path of the vehicle. The IvP Helm will decide on a compromise between the two vehicles in order to avoid the obstacle while still staying on track to reach the waypoint.

IvP Helm Functionality

The IvP Helm structure and functionality can be seen in Figure 13. The IvP Helm is a MOOSApp and so it is connected to a MOOSDB where it receives information as input and outputs its autonomous decisions to. The IvP Helm manages all of the behaviors which are being used in a mission. Based on the information it receives from the MOOSDB, it will determine which behaviors should be active at any given time during mission execution. The IvP Helm will then execute the functionality of all of the active behaviors. A behavior will examine the current state of the mission and produce what is known as an IvP Function. IvP Functions are mathematical functions which are used by the IvP Helm to derive the actual final decision for the given iterate loop.

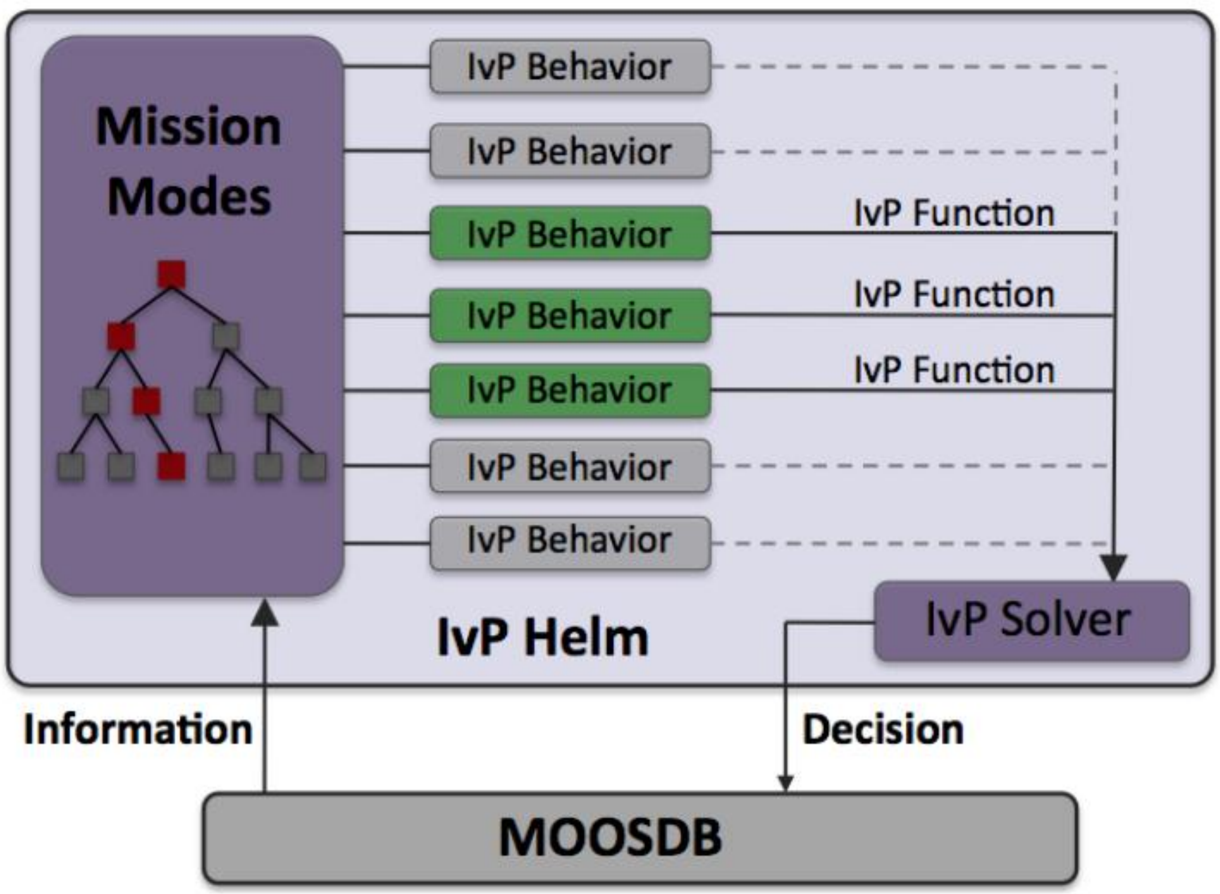


FIGURE 13: THE IVP HELM INTERNAL STRUCTURE. [2]

MOOS-IvP Missions

MOOS-IvP missions are merely a set of configuration files. There are two types of configuration files which MOOS-IvP uses, one for MOOSApps and one for behaviors. Through the use of the configuration files, the user tells MOOS-IvP which MOOSApps and behaviors to use and how those applications should function. The two types of configuration files are similar in structure and control various parameters that the different programs need to operate in the desired way for the mission. A mission is launched using an application that comes with MOOS-IvP called pAntler. The pAntler application takes all of the configuration files for the mission as arguments and launches them with a new MOOSDB. Behaviors are loaded directly through the configuration block for the IvP Helm so when the IvP Helm is started, it loads the specified behaviors and their configurations.

Programming MOOS-IvP

Because MOOS-IvP is open source, it can be modified and expanded upon to meet users' needs. Developers can write their own MOOSApps as well as new behaviors if the applications provided with MOOS-IvP or the readily available third-party applications do not suit the user's needs. Since MOOS is written in C++, new MOOSApps and behaviors merely need to extend their respective built-in classes and overwrite key methods which produce the functionality of an individual application.

Since the behaviors provided by the core MOOS-IvP package were sufficient for the goals of the ASV project, no new behaviors had to be written. However, MOOS-IvP does not come with an interface for transferring autonomy information from MOOS-IvP to the controls of the vehicle, known as a payload autonomy interface, since such an interface is application specific. Therefore, a payload autonomy interface had to be written for sending values output from the IvP Helm to the controls of the vehicle and receiving back sensory information to be posted to the MOOSDB.

Payload Autonomy Interface

The payload autonomy interface is a MOOSApp which subscribes to MOOSDB variables that are published by the IvP Helm. It then forwards this data to the Arduino over the USB serial connection, as seen in Figure 14. At the same time, the interface reads in sensor data sent from the Arduino and forwards this data to the MOOSDB to be read by the IvP Helm. The data sent to the Arduino includes

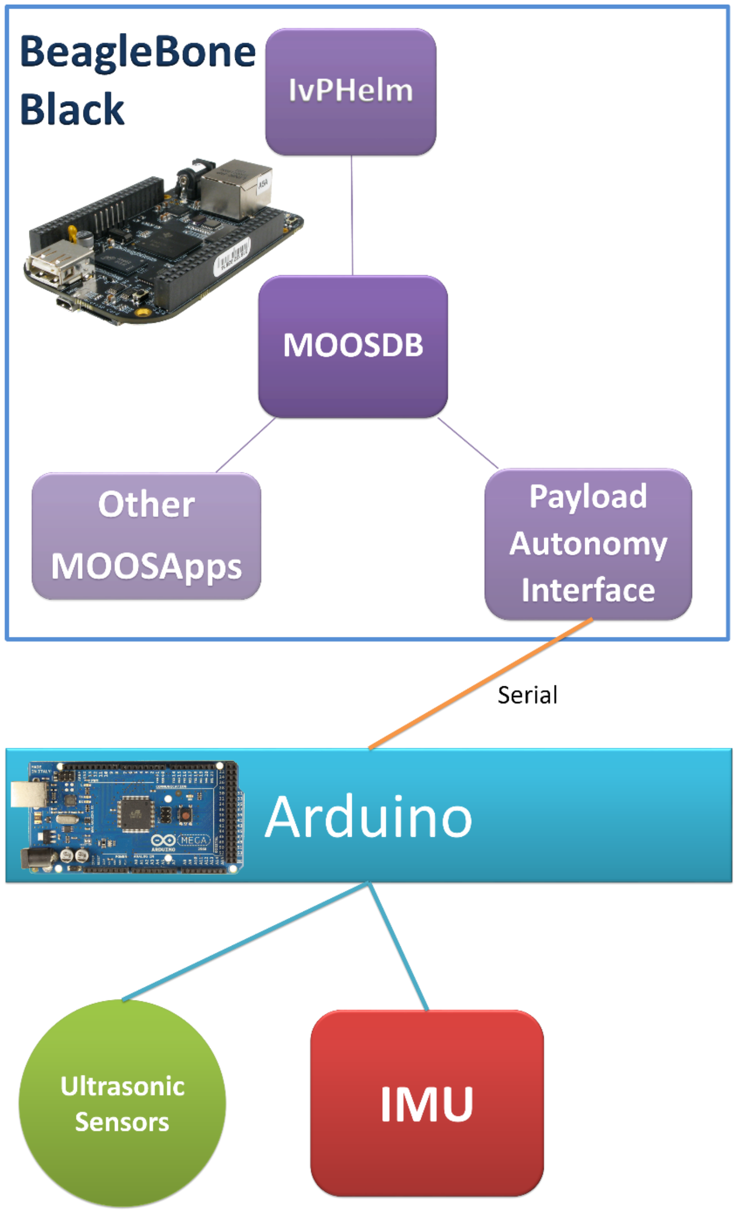


FIGURE 14: OVERVIEW OF THE MAIN SOFTWARE COMPONENTS AND THEIR RELATION TO THE HARDWARE THEY ARE CONTAINED IN AND INTERFACE WITH.

desired heading and speed. Data can then be sent back from the Arduino in the form of position, actual heading and speed, and detected obstacle data. This data flow can be seen in Figure 15.

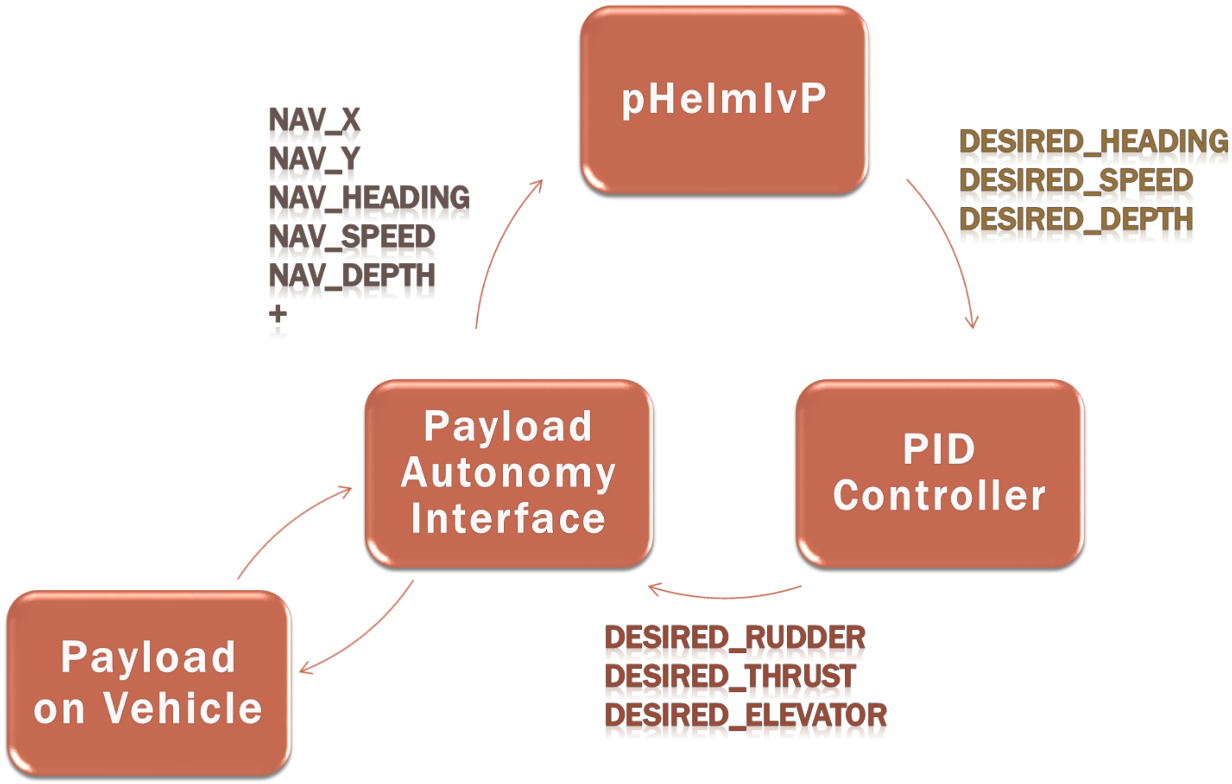


FIGURE 15: THE DATA FLOW TO AND FROM THE PAYLOAD AUTONOMY INTERFACE. THE ACTUAL DATA FLOW WITHIN MOOS-IVP GOES THROUGH THE MOOSDB. [2]

Messages sent to and from the Arduino are in a *key=value* form where the *key* is the name of the type of message being sent and the *value* is the actual data. This format was chosen over others because of its human readability and it works especially well since there is only a small set of keys that are needed on each end of the communication. These key-value pairs are separated by a delimiter in order to be able to be distinguish between different messages buffered in the serial interfaces. The default delimiter character is a comma but the messaging system was programmed in such a way that it can be easily changed.

With the payload autonomy interface setup, data is seamlessly passed to and from the frontend on the vehicle thus creating a contained loop of information necessary for the autonomous operations of the vehicle.

Feedback Systems

Purpose

For autonomy to function efficiently and accurately, a precise feedback system is necessary. As MOOS-IvP runs autonomy, it requires specific parameters about the boat itself and its environment in order to make precise decisions on how to direct the vehicle. The main parameters it needs are:

- Obstacles (When necessary)
- Heading
- Speed
- Location

Obstacle Avoidance

Almost all successful autonomous vehicles require some sort of system that reaches out to the environment of potential risks or obstructions. To achieve this above water, an ultrasonic sensor is used. In deciding on which would fit the needs of the ASV, distance had to be taken account. For the purpose of testing logic, speed was not a goal for the ASV. With this, the ASV is to be controlled at a slower speed to increase the amount of time MOOS-IvP had to react to any potential objects. With this, we would want a sensor that could pick up an obstacle as early as possible. Assuming the ASV was traveling no more than 0.25 m/s, for every meter an object was away, the boat had about four seconds before hitting an object. With what is available in the market, a HC-SR04 Ultrasonic Ranging Module was chosen, as seen in Figure 16.

FIGURE 16: HR-S04 ULTRASONIC RANGING MODULE

The sensor works by using two microphones: one for ‘pinging’ (trigger) and the other for receiving the reflected sent ping (echo). This pinging is a process of sending an ultrasonic wave in one direction, at a speed of 340m/s, while timing how long it takes for it to come back. When a ping meets an object, it will then bounce back and be received by the receiving microphone.

Using this measured time, the distance the said object away can then be calculated by Equation 1 and shown in Figure 17.

$$Distance = \frac{(ultrasonic\ spread\ velocity)(timer\ record)}{2} \qquad (1)$$

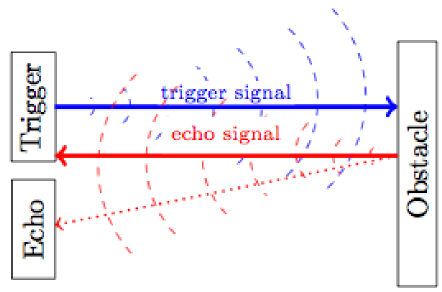


FIGURE 17 PROCESS OF PINGING FOR OBSTACLE WITH AN ULTRASONIC RANGING MODULE. [1]

The HC-SR04 has limitations however of what is can measure. This particular sensor has a measuring angle of 30° and a measuring distance between 0.02-4meters. With this, the sensor is also limited at what kind of objects it can pick up. As the sensor is pinging, it is assumed the obstacle is situated so that it has a surface relatively facing the sensor. However, if the object is turned at an angle greater than 30°, the echo signal will not be picked up by the receiver, thus causing unusual measurements.

To resolve issues such as these, the microprocessor running the sensor has coding in it to recognize when this happens and to assume that an object is there for the time being, to reduce the risk. With the sensor operating at an ultrasonic frequency of 40 kHz, it has the potential to produce large amounts of data. Forwarding all this data could potentially overload the onboard computer. The microprocessors therefore, are coded to filter out any distances measured over three meters as anything over that would be considered not a risk to the ASV.

By increasing the amount of sensors on the ASV, we effectively increase the vision and thus the accuracy of obstacle avoidance. When setting up the sensors, the angle of its positioning has to be taken into account. Ideally, three sensors should be used. For the case of the ASV, four are used to increase accuracy.

All sensors need to be positioned so that all angles are at with one another, with an ideal radius centered in reference to a specific point behind them, as shown in Figure 18.

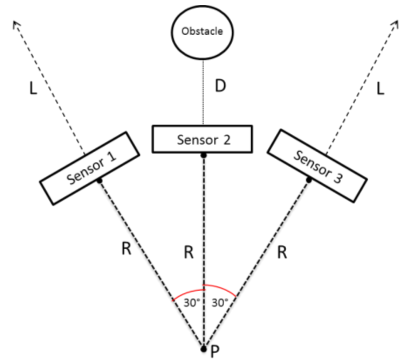


FIGURE 18: SETUP OF THREE-SENSOR SYSTEM APPROACHING OBSTACLE.

As the ASV approaches an obstacle and read a distance less than its risk distance, 'L' (3 meters), it will learn that there is an object ahead, shown in Figure 19, as distance 'D'. It will compare that distance with that of the distance readings of the other sensors. If that is the only object, it will learn that all the other sensors are reading a distance of 'L' and therefore will let MOOS-IvP know that there is one object away

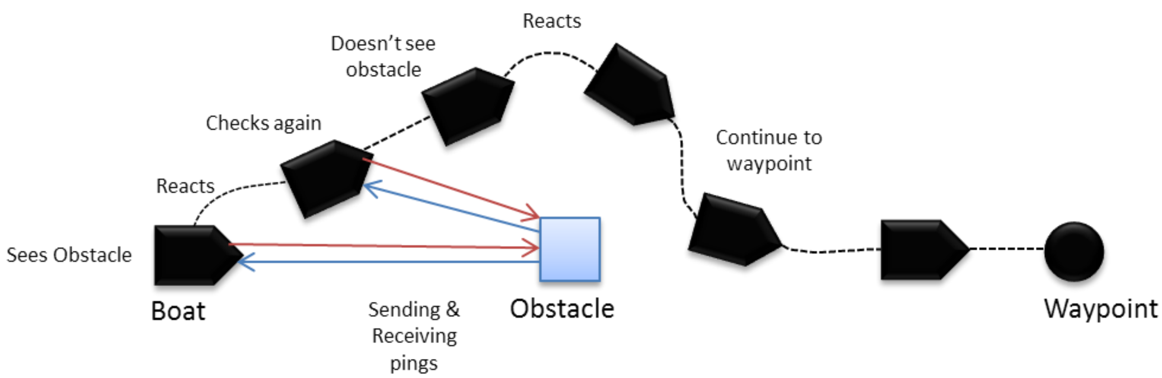


FIGURE 19: DIAGRAM SHOWING ASV LOGIC APPROACHING OBSTACLE.

at distance 'L'. From there the boat will be told to either go left or right, and continue to ping until it has past the object and continue on its way.

Heading and Positioning

MOOS-IvP requires an input parameter of its position relative to the area, in order to for it to follow course to its specified waypoint. With testing an ASV indoors, experimentation is close to impossible with the use of a GPS. Indoors, GPS cannot accurately position itself due to the lack of communication with the satellites. With typical experiments, the area of testing will usually be relatively small compared to the distance between the GPS module and the satellites being used, creating a large area for error in positioning. Another method of positioning without the use of communicating with outside components (as GPS uses satellites) is then required to a method referred to as Dead Reckoning. For testing, this method should be used in limited timed trials as over time, the small error of positioning will propagate due to this method's restriction in accuracy. Therefore, this needs to be reset every time a new trial is to be started.

To complete Dead Reckoning, an IMU (Inertial Measurement Unit) is needed for use to report on the ASV's velocity, orientation and gravitational forces. This is done by the IMU's three main components: three accelerometers, three gyroscopes, and three magnetometer, making up its nine degrees of freedom.

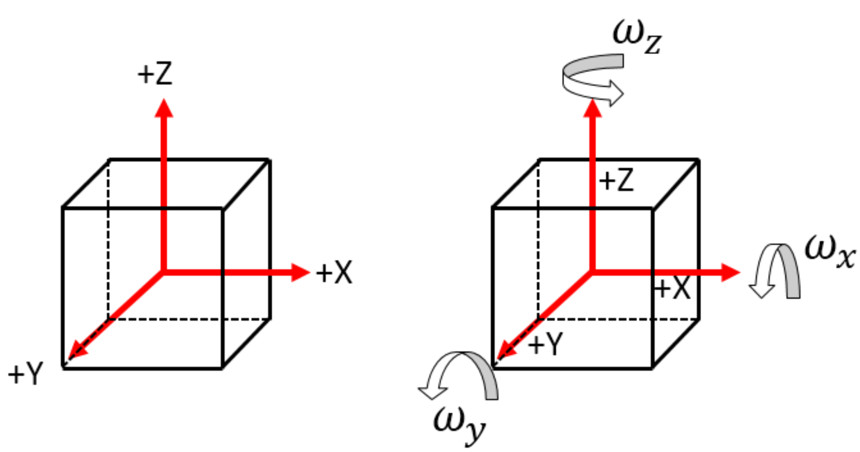


Figure 20: (Left) Displays the axis of three accelerometer. (Right) Displays the axis of motion of gyroscope.

By using the accelerometer and gyroscope, the three degrees of freedom of each component (x, y, z *and* $\theta_x, \theta_y, \theta_z$ respectively) and integrates over time the measured acceleration, with the estimation of gravity, to then calculate the velocity. This axis reference can be seen in Figure 20. From the velocity, it then will integrate it to calculate the current position, relative to the original starting point. The magnetometer is essentially an onboard compass that uses true north heading to calculate which direction it is moving. Figure 21 shows how all of these components work together.

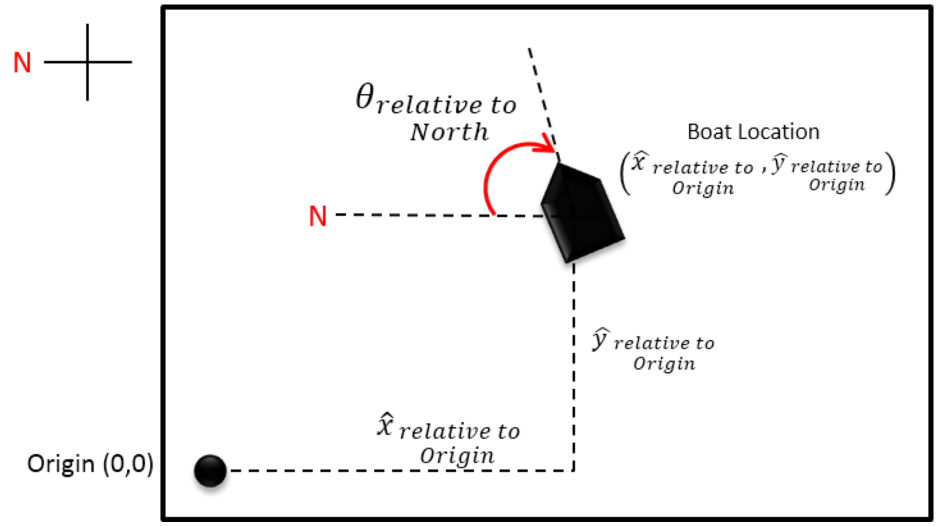


FIGURE 21: DIAGRAM DISPLAYING HOW THE ASV WILL USE DEAD RECKONING TO LEARN ITS POSITION RELATIVE TO THE ORIGIN, HEADING, AND VELOCITY.

Future Work

Immediate

At the time of writing this report there are no final experimental results to account for due to time constraints and manufacturer issues (we were given a few wrong parts). This does not mean that by the time of the presentation of the project we do not hope to have final results though. Once all of the required parts have come in the group intends to test for autonomy. First the ASV will be run purely as a remote control vehicle to make sure all of the physical systems are working correctly. After the ASV will be wired to be both autonomous and remote controlled and tasks 2 and 3 to prove autonomy. All of the subsystems of the ASV have been tested and function alone so implementing them all into one system is the only remaining step, and is expected to be done by 6 May 2014.

Long-Term

Beyond 6 May 2014 (end of the UNH 2013-2014 Spring Term) the ASV project has been funded to continue working over the summer. NEEC (Naval Engineering Education Center) has provided a grant for this continued work and after proof of concept from the RC sized prototype would like a higher integrity model to be developed. This high integrity prototype is to be approximately the size of a RHIB (rigid hull inflatable boat) or a canoe. The high integrity prototype is being developed with the intention of testing outside as opposed to indoors solving many issues. Because the high integrity prototype is outside GPS can be used to solve the positioning issue the group ran into. Also the budget for this model is much higher, and as such higher end equipment and sensors can be used which is expected to significantly increase the accuracy of the autonomy as well as the overall speed at which the system can operate. Beyond this the group hopes to receive more funding from NEEC to continue to project further and make a fleet of ASVs that can communicate amongst themselves to further increase their operating speed and efficiency.

RESULTS AND DISCUSSIONS

Due to the nature of the project and time constraints, at the time of writing the report there are no final experimental results to discuss. However at the time of presenting the group hopes to have conclusive experimental data of the entire system. While the system as a whole has yet to be tested, each subsystem of the ASV has been tested alone and functions properly. Each motor, ESC, servo combination has been tested using remote control inputs and functions as expected. The starting torque of the chosen motors is high, but this is expected due to their size. This is not an issue in the initial prototype as there is more than enough power to overcome the initial torque barrier, but it is something to consider as to project progresses and requires larger motors. Finding the calibration for the encoders (RPM of motors to actual speed of boat) is difficult, however the discrepancy in conversion can be diminished once the project progresses outside and GPS can be used as a secondary source of feedback for speed. From testing the ultrasonic sensors individually it is determined that there maximum range is only about 3m, while this is sufficient for the small scale model as the project progresses to a larger scale the obstacle avoidance sensors will need a much larger range. Due to having a higher budget for the higher integrity prototype higher grade, longer range ultrasonic sensors can be used. Similarly testing with the PlayStation®Eye camera has shown the range to be limited to about 20”. The goal for the higher integrity model is to use similar visual recognition software/programming but have higher end cameras with greater resolution.

Conclusion

With no experimental results of the entire system at the time of writing this report it is hard to come to a conclusion, however the goal is that by performing the three simple autonomous tasks assigned to the project the group can prove autonomy and progress to a higher integrity model that will allow for proof of concept of even greater autonomy.

REFERENCES

[1] Formenti, n. (2013, November 16). *HC-SR04: ultrasonic sensor for arduino.* Retrieved from Macduino Blogspot: http://macduino.blogspot.com/2013/11/HC-SR04-part1.html

[2] Michael R. Benjamin, H. S. (2013, May 10). *An Overview of MOOS-IvP and User Guide to the IvP Helm - Release 13.5.* Retrieved from MIT: http://oceanai.mit.edu/moos-ivp-pdf/moosivp-helm.pdf

[3] Sanz, P. J. (2013, April 16). *Marine Robots and Dexterous Manipulation for Enabling Autonomous Underwater Multipurpose Intervention Mission.* Retrieved from Trident: http://www.irs.uji.es/trident/flyer.pdf

APPENDICES

Appendix A

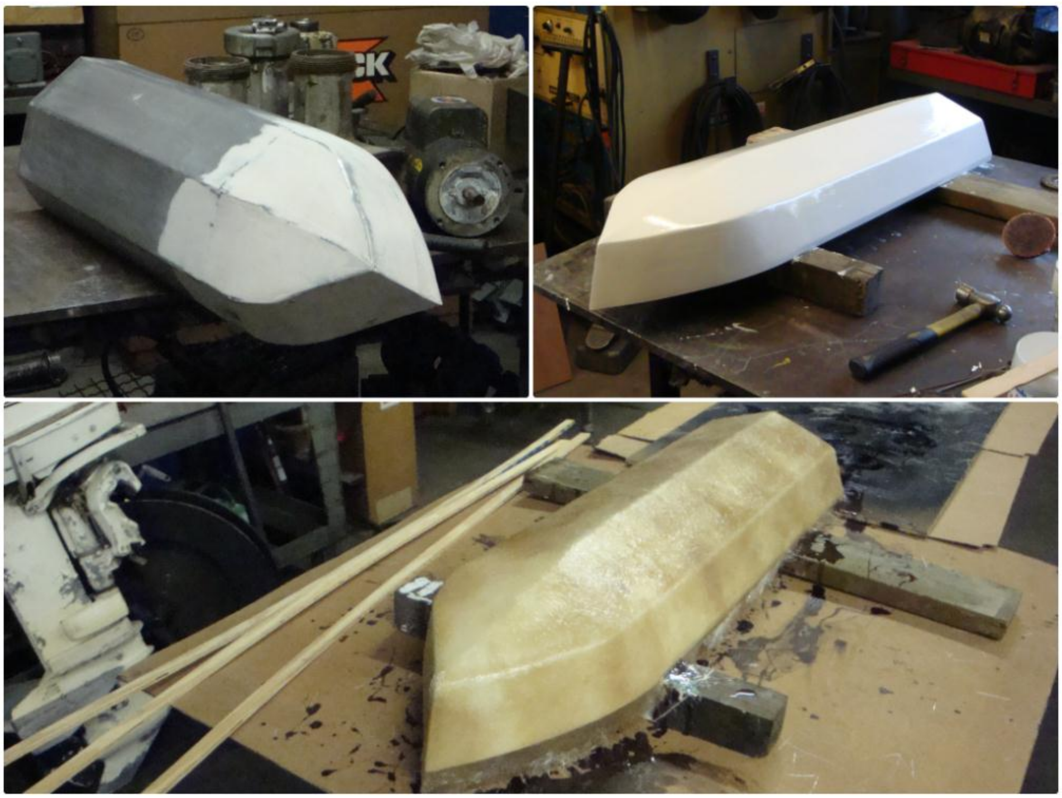


FIGURE 22: THE FIRST MOLD, THE FIRST MOLD WITH A SMOOTH PUTTY FINISH APPLIED AND THEN THE FIBER GLASS WHICH WAS PLACED OVER THE TOLD TO CREATE THE SECOND MOLD.

FIGURE 23: THE SECOND MOLD OF THE BOTTOM OF THE HULL. THE SURROUNDING CARDBOARD WAS IMPLEMENTED IN ORDER TO MAINTAIN THE STRUCTURAL INTEGRITY OF THE HULL DURING FIBER GLASSING AS IT TEND TO SHRINK AS IT DRIES.