Design and Realization of an Autonomous Underwater Vehicle: Max'inux

Simon Honigmann*, Ryan Murch*, Gareth Ellis[⸸], Cameron Newton[†], Viral Galaiya[‡], Joel Ahn[†], Vincent Yuan[‡]
*Department of Mechanical Engineering, University of British Columbia, Vancouver, Canada
[⸸]Department of Computer Science, University of British Columbia, Vancouver, Canada
[†]Department of Engineering Physics, University of British Columbia, Vancouver, Canada
[‡]Department of Electrical and Computer Engineering, University of British Columbia, Vancouver, Canada

***Abstract*—Max'inux (Max), is the University of British Columbia's (UBC) Subbots team's inaugural autonomous underwater vehicle (AUV), developed for the 2018 AUVSI Robosub competition. Max was designed and built by a diverse team of undergraduate students at the University of British Columbia to be robust, reconfigurable, and to serve as a foundation for future teams to continue to develop and learn from. The vehicle implements a novel under-actuated control scheme and a highly optimized linear quadratic regulator (LQR) controller. The software uses a world state architecture to quickly allow new processes, interactions, and objectives to be added to the vehicles list of capabilities. Max marks Subbots' first foray into underwater robotics and the beginning of a long journey exploring the field.**

***Keywords*—autonomous underwater vehicle, RoboSub 2018**

1. INTRODUCTION

The Subbots team was founded in 2017 as a joint venture between UBC's Human Powered Submarine Team and Autonomous Ground Vehicles Team. The team is entirely comprised of student volunteers from various engineering departments and years of study at UBC who must fund, design, and construct the project themselves. This paper documents the conception of the team's first AUV. The vehicle was named Max'inux to pay tribute to the local aboriginal Canadian groups, on whose ancestral lands UBC was built. Max'inux (Max) translates to killer whale, a symbol of the Pacific Northwest from where the team hails. Max is shown in the rendering below.

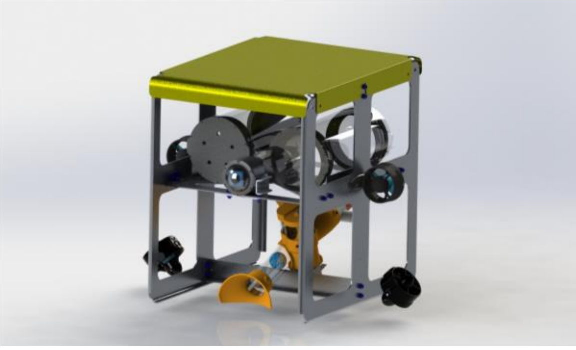

Fig. 1: Max'inux – Subbots' First AUV

2. VEHICLE SPECIFICATIONS

The following table outlines Max's key parameters and specifications:

TABLE 1: Max'inux Vehicle Specifications

Dimensions	
Mass	
Computation	Intel NUC 7th Gen Intel Core i5 8 GB DDR4 RAM 64GB M2 SSD
Software Architecture	Robot Operating System (ROS)
Power	Venom 8000mAh 3S15C LiPo Venom 5000mAh 3S20C LiPo
Vision	(2x) ELP 1080p Fisheye Camera
Navigation	Phidgets 1042 IMU (2x) TE501 Sonar Depth Sensor
Propulsion	(2x) Blue Robotics T200 Thrusters (2x) Blue Robotics T100 Thrusters

3. COMPETITION STRATEGY

As the newest group competing at the 2018 RoboSub competition, developing a strategy compatible with the team's schedule, experience, and resources was paramount to project success. With limited man hours to dedicate to the project and much ground to make up, the team decided to focus its efforts towards reliably completing basic operations and to apply the lessons learned to more ambitious goals in future years. The minimum requirement for the competition involves identifying and moving through a gate. As such, the focus was on developing robust navigation and object recognition, which could then be reapplied, with no additional hardware, to other objectives.

From the mechanical engineering perspective, this led to a focus on designing a reconfigurable frame taking a design-for-manufacture approach. All designed components, mounts, and actuators were made such that they could be manufactured using a water jet cutter, 3D printers, or standard machine tools. New mounts can be readily installed on the frame without compromising its integrity allowing for changes to be made at a moment's notice. Vehicle size, weight, and enclosure complexity were all reduced by opting for an under-actuated thruster scheme.

Software was built to be adaptable and expandable. Focused on loose coupling of submodules to allow for smooth updates. Separated ROS implementation from algorithmic complexity to allow for multiple processing changes without affecting the functionality. A model-centric development strategy was used to keep development costs and risks relatively low, and to allow the software team to test algorithms prior to the vehicle's construction. This strategy included using Solidworks simulations to test structural integrity and estimate vehicle fluid dynamics, as well as using UWSim to quickly test vehicle control algorithms.

The team's ultimate goal is to provide unfettered learning experience for its members, to better prepare members for industry, and to leave members comfortable independently designing and testing systems. To achieve this goal, the team attempted to build as many things in house as possible, including testing piezoelectric transducers for sonar, and thermoforming custom viewports. This helps members gain a better understanding of what works and what doesn't, gives them improved ability to tailor designs to needs, and reduces their reliance on costly and slow external sources. Finally, by making attending competition the team's top priority, the team ensures that members will be able to learn from and network with other teams and industry leaders.

4. DESIGN CREATIVITY

One of the first decisions the team made was to select an underactuated control scheme, inspired by the RRC ROV II.[1] The team was drawn to the idea for several reasons including the reduced cost and weight of having fewer actuators, the added challenge in developing an optimal controller, and the novelty of the configuration. This selection shaped requirements for the electrical system, vehicle frame, and system software. Details for each will be discussed in the following sections.

4.1 Frame and Enclosures

The approach to the frame and enclosure design was to optimize rigidity while minimizing weight and ensuring that the design is modular, adaptive, and rapidly-prototypable. The frame was scheduled to be the first completed system, to give the other sub-teams a platform on which to develop their respective designs. A detailed material comparison was performed using CES Selector 2017, where stiffness and strength were compared to density. The top 3 materials suggested by the software, in order, were carbon fiber composite, aluminum, and glass fiber composite. Ultimately, due to the team's relative inexperience with composite material fabrication, the relatively high cost of composites vs. traditional machining, and the desire for reconfigurability, the team selected aluminum as the primary material for the frame.

The team then researched existing ROVs and AUVs to draw inspiration for successful designs, and to learn from the pitfalls of other vehicles. A traditional plate truss frame design was selected as it best satisfied the criteria of modularity, design for manufacture, and adaptability. Solidworks was used to perform both static and dynamic load simulations on prospective designs. Over the course of over 20 designs and modifications, the plate and crossbeam structure used on the vehicle was improved, to reduce weight. The frame was designed to be easily waterjet cut and bent in-house by team members. Aluminum angle stock was added for structural support. With a final mass of 4kg, the frame is designed to comfortably handle a payload of 20kg.

Component placement can be readily modified by drilling new mount points directly into the plates. The center or side plates can also be easily replaced, in the event of a changed horizontal or vertical profile requirement. The frame, while being structurally sound, leaves two large unrestricted areas for enclosures and actuators to be mounted. The size of the frame allows for ballast and buoyancy foam to be placed in such a way that stability is improved - a key consideration given the vehicle's underactuated control scheme discussed in the following section.

4.2 Actuator Design

Initially design decisions were undertaken based off the assumption of the previous year's competition. From these assumptions the sub-team conducted function decomposition and to identify tasks. The team decided to continue with idea generation before and after doing market research to allow for both proven and novel ideas.

System priority was decided based on which components were most necessary to score the most points. The team identified the ball collection and manipulation as the most important feature. The ball collection design consists of two tasks: ball acquisition and ball storage.

4.2.1 Ball Collection:

For the acquisition portion we decided that using a pump would be the best method of collecting balls as it naturally pulls the balls to the inlet and translates them to storage, while remaining extremely simple. This significantly reduces mechanical complexity, saving design and manufacturing time. Commercial bilge pumps were selected to take advantage of their inherent waterproofing and low cost. Multiple pumps are coupled together to allow for faster ball collection and a wider area of influence. This allows more error tolerance for position of the overall robot when collecting balls. The CAD assembly of the ball collection and storage unit can be seen below.

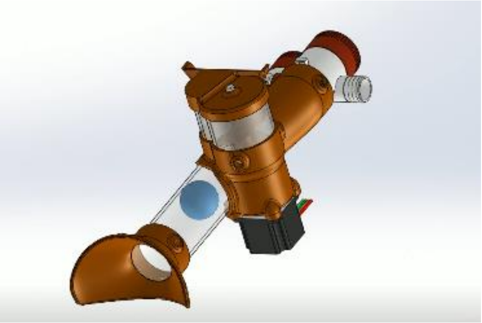

Fig. 2: Golf Ball Collection System

4.2.2 Ball Storage:

Out of the several systems that were considered, an Archimedes screw was selected as the most suitable to move balls outside of the main acquisition tube. This addition increases the vehicle's capacity for both storage and retrieval. The screw design has the added benefit of being able to raise balls to the top of the vehicle for future objectives. The screw pushes the balls up out of the flow and into the ball storage area. When the screw rotates backwards the balls can descend into the acquisition tube and gravity allows it to fall out of the hole.

4.3 Controller Design

The Linear Quadratic regulator (LQR) was implemented to control the heave position and the velocity of pitch, roll, yaw, sway and surge. The decision to use the LQR opposed to the more conventional proportional integral derivative (PID) was to minimize the energy based cost of the thruster. LQR also enables the provision of weights to optimize the more important states providing a better response for the selected states.[2] The controller is based on a linearized and simplified model. The linear position, angular position, linear velocity and angular velocity were the 12 states of the model. The model was created on MATLAB and tuned by simulating the response on Simulink.

4.4 Software Architecture

The software of the AUV is built using ROS and various ROS terminology will be used throughout the section. The design is composed of 4 main components: input, world state, decision and control.

At a high level, the various inputs are measurements of the world around AUV and fed to the state and decision nodes. The world state node is a state machine that uses input to decide what the current task is and monitors for completion criteria to move on to the next task, the current task is the state and this is published out to the decision node. The decision node uses the input and the current state to make movement decisions, which are then sent to motor control.

4.4.1 *Input:*

The main inputs for our AUV are 2 cameras, one front-facing and one down-facing. Other inputs used include an IMU, depth sensor and a hydrophone array. Camera feeds are sent through an HSV filter corresponding to the specific needs of the current task to identify key features. These features are then detected by a node and published as discrete data. For example, for a task of following an orange line the camera would take frames of the line, those frames would be filtered to just show orange, and then a node would detect the direction of the line and publish a message saying it sees a line and give angle of the line.

4.4.2 *World State:*

The world state node is a state machine that broadcasts a message containing the current state, which is the current task we are trying to accomplish. The node's architecture is similar to that of a master-slave style where only one slave is operating at a time. The slaves of the world state node are called routines. The master will start up a slave corresponding to the present task. The slave will then monitor the inputs for criteria indicating a state change and will notify the master of that detection, at which point the master will tell the slave to stop and start up a new one responsible for the new task/state.

4.4.3 *Decision:*

The decision node is responsible for providing movement instructions to the control node based on the current state and inputs. This node has a very similar architecture as the world state node, a master-slave relationship with only one active slave. The slaves of the decision node are called Subroutines. In this case the decision node subscribes to the world state and publishes movement directions. Upon a state change the decision node stops the current running subroutine and starts a new one. Each subroutine is catered to a specific task and knows what inputs it should consider and how to react to them. It then informs the control node of its decision.

4.4.4 *Control:*

Decisions are passed to the control node as a combination of position and velocity inputs. The role of the control node is two-fold: first it develops a vehicle state estimate using probabilistic sensor fusion. Then, the control node acts on commands from the decision node using the discrete LQR controller discussed in the previous section. A PWM value is sent to the vehicle firmware to directly control thruster speed, to optimally execute navigation commands.

5. Experimental Results

5.1 Software Testing

There are three principal strategies for rigorously testing the algorithms: ROS unit tests, UWSim simulation tests, and physical tests on the robot. Both physical and simulated models of the RoboSub competition challenges, such as the gate and the dice, have been constructed. Simulated images, like the one shown below, helped the team test navigation and object detection algorithms early on.

Fig 2: UWSim simulation in progress

In the simulator, however, only the system's ability to handle camera data is being tested, without any of the corroboration of other sensors such as an IMU or hydrophone. The most important testing milestone will come from our physical integration tests, where the parameters governing the robot's algorithms are most finely tuned.

5.2 Dry Enclosure Testing

All enclosures and cables tested using a 3-part test approach. First, enclosures and cables are placed in a bucket of water and monitored to check for fast leaks. After passing the fast leak check, the next stage is vacuum testing. By connecting a vacuum pump to one side of an enclosure or cable gland and submerging the enclosure in water, a pressure difference of 1atm can be simulated without requiring a pressure pot or a 10m water column. This method has worked remarkably well, as leaks can be easily spotted through the clear acrylic enclosures by the characteristic sputtering bubbles as the water quickly vaporizes in the vacuum. Once a component has passed the vacuum test, the final test performed is prolonged submersion at depth. The enclosure is submerged at the bottom of a pool overnight to ensure that no slow leaks are present.

5.3 Vehicle Testing

For the purpose of system testing and vehicle testing, a large test tank was constructed. At 3m long, 1.5m wide, and 1.2m deep, the tank allows for limited but reasonable movement of the vehicle, and interaction with scaled replicas of competition obstacles. The tank allows basic navigation, vision, and stability tests to be performed at a moment's notice. For more involved testing, the UBC Aquatic Centre is used. Full scale replicas of the qualification gate and dice objectives were constructed to test the vehicle's software.

6. CONCLUSION

Max'inux is a robust autonomous underwater vehicle, designed to be a low cost, reconfigurable test platform for future designs. Max can effectively identify objects and objectives underwater, and navigate through and around obstacles. Further work will improve on Max's hydrodynamics, add increased capacity to perform different tasks, and optimize systems based on the team's competition experience.

7. ACKNOWLEDGMENT

The Subbots team would first and foremost like to thank its hard-working student volunteers who commit their weekends and free time (a scarce commodity for engineering students) to working on the team. It is a joy to work with a team that is so passionate and committed to this project.

The team would also like to thank the generous support of the Departments of Mechanical Engineering and Electrical Engineering, as well as the Faculty of Applied Science. UBC's engineering faculty and staff, including our faculty advisor, Prof. Jon Mikkelsen, and our machine shop supervisor, Markus Fengler, have been incredibly helpful in helping the team get off the ground.

The team would like to thank all of its external sources of support, including: EGBC, IEEE, SNAME, Misumi, Shell Canada, Dassault Systemes, MathWorks, Phidgets, Spaenaur, Venom, Viking Profiles, and 4Imprint Canada. The success of the project is contingent on generous support from sponsors, and the team would not be able to offer such a rich learning experience for students without their help.

Finally, the team would like to express its sincerest gratitude to the Robosub competition organizers and volunteers for working tirelessly to put on such a wonderful event, and the US Navy for letting Robosub borrow their wonderful facilities to promote international learning. Despite being the new kids on the block (with silly Canadian accents), the team has found the Robosub community to be wonderfully supportive and was happy to become a part of it.

References

[1] C. S. Chin. “A Robust Controller Design Method and Stability Analysis of an Underactuated Underwater Vehicle”. 2006.

[2] P. C. Young and J. C. Willems, “An approach to the linear multivariable servomechanism problem”, International Journal of Control, Volume 15, Issue 5, 1972, pp. 961-979.

Appendix A: Component Specs

Component	Vendor	Model/Type	Specs
Bilge pump	Amazon	Sailflo	1100 GPH
Polycarbonate Tube 2.00” OD	McMaster Carr		⅛” walls 1’
Polycarbonate Tube 2.25” OD	McMaster Carr		⅛” walls 2’
Polycarbonate Tube 2.75” OD	McMaster Carr		⅛” walls 1’
Brass heat set inserts ¼”	McMaster Carr		
Battery	Venom Power		8000mAh 3S 11.1V 15C LiPo
Battery	Venom Power		5000mAh 3S 11.1V 20C LiPo
Thruster	Blue Robotics	T-200	
Thruster	Blue Robotics	T-100	
Electronic Speed Controllers	Blue Robotics		30A
Arduino Mega 2560	Amazon		
On-board Computer	Newegg	Intel NUC	
Relay Module	Digikey	Elegoo	4 Channel, 5VDC
Arduino Shield	Amazon		
IP68 Cable Gland	Digikey	Amphenol	0.12” - 0.26” diameter, M12x1.5 thread size, nylon
Female Cable Connector	Digikey	EN3	IP68, 3 position
Male Cable Connector	Digikey	EN3	IP68, 3 position
Female Cable Connector	Digikey	EN3	IP68, 4 position
Male Cable Connector	Digikey	EN3	IP68, 4 position
4 cond. 22 AWG Cable	Digikey	Alpha Wire	PVC Insulated
3 cond. 18 AWG Cable	Digikey	Alpha Wire	PVC Insulated
Through Hole Relay	Digikey	Panasonic	12VDC, 30A, SPST
Through Hole Terminal Blocks	Digikey	Phoenix	500V, 30A, 2 position