Proceedings of the Fifteenth Annual Early Career Technical Conference
The University of Alabama, Birmingham ECTC 2015
November 7 – 8, 2015 - Birmingham, Alabama USA

AZI-DRIVE: A NOVEL THRUSTER MAPPER FOR TWIN PROPELLER MARITIME VESSELS

Andrew Gray
University of Florida
Gainesville, Florida, U.S.A

Kevin French
University of Florida
Gainesville, Florida, U.S.A.

Jacob Panikulam
University of Florida
Gainesville, Florida, U.S.A

Zach Goins
University of Florida
Gainesville, Florida, U.S.A

Dr. Eric Schwartz
University of Florida
Gainesville, Florida, U.S.A

ABSTRACT

Over-actuated surface vessels, while highly maneuverable, can be challenging to control. This paper describes the process of implementing the controls of a novel propulsion system onto a small, fully autonomous surface vehicle (ASV). Two independent azimuth steered rimless propellers provide propulsion for the vessel. Using a nonlinear control allocation with singularity avoidance method, the approach was tested with simulations and then implemented on an ASV. Using the singularity avoidance variable, users could adjust the responsiveness of the system. Testing confirmed that the method allowed the ASV to move in all three dimensions required of a surface vehicle.

INTRODUCTION

For most maritime vessels, a propeller and rudder are adequate to allow a boat to transit from one point to another. However, for complex maneuvers such as station holding performed by dynamic positioning (DP) vessels, more complicated propulsion systems are required. These systems are generally over-actuated [1] and require complex control software to utilize the thrusters.

Students from the Machine Intelligence Lab (MIL) at the University of Florida have a history of developing autonomous maritime vessel [2][3][4][5]. PropaGator 2, MIL's most recent ASV shown in Figure 1, was designed to have a top speed of 10 knots while maintaining the maneuverability of a DP vessel [6]. To accomplish these characteristics, a planing hull was developed with a low drag coefficient in the forward direction. For propulsion, two student-designed azimuth thrusters were installed on the back half of the vessel.

While the hull design and propulsion system on PropaGator 2 allowed for high speeds, the complexity of the system created a challenging control problem. In simulation, the two thrusters were able to control the boat simultaneously in all three dimensions [7], but implementation proved difficult. Additionally, the placement of the thrusters at the rear combined with large drag resistance across the perpendicular plane made strafing a challenge. The thrusters produce more thrust in the forward direction than in the reverse. While the drag of the propellers is very small, the drag force experienced by the boat is far greater in the reverse than in the forward direction. Another challenge was that due to the plowing characteristics of the boat, speed must be considered in its control.

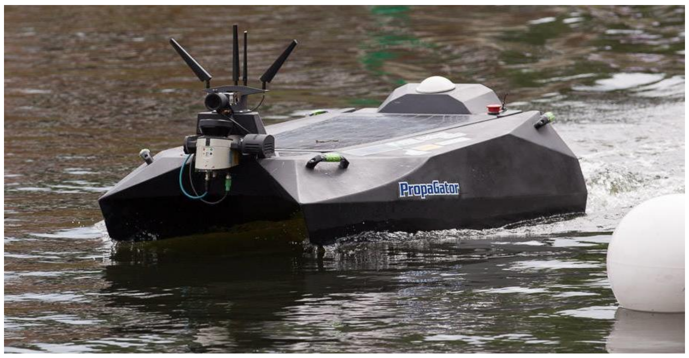
Figure 1. The autonomous boat PropaGator 2

The propulsion system on PropaGator 2 does not fit the model of a traditional DP ship. First, PropaGator 2 is only six feet long. Most DP vessels are meant for the open ocean and are hundreds of feet long. PropaGator 2 is only able to operate in small bodies of water with only limited waves. Traditional DP ships will have two to three times more thrusters than PropaGator 2. Numerous thrusters allow for redundancy and can make the vessel's movements more responsive. With only two azimuth thrusters, the ASV cannot afford to lose a thruster; in some cases, the ASV may be less responsive than a vessel with four or five thrusters.

To control the propulsion system on PropaGator 2, a student implemented constrained nonlinear control allocation with singularity avoidance thruster mapper was used. The thruster mapper allowed the ASV to move in any direction while maintaining the ability to hold its position and heading. This ability classified PropaGator 2 as a DP vessel and was proven by testing the platform in several small lakes.

2. PLATFORM SUMMARY

PropaGator 2 was made to be fast, maneuverable, and lightweight. A catamaran hull was selected as the hull type because catamaran hulls typically produce less drag when compared to a similar sized monohull [8]. Hard chines were added to the bottom of the vessel to allow for a planing hull using previously conducted hydrofoil research [6]. A hard transom was installed at the back of the boat. The hard transom caused the water to cleanly separate from the boat in order to reduced drag. Fiberglass was selected for the hull material because of its strength and low weight. For propulsion, two rim driven azimuth thrusters, shown in Figure 2, were installed near the stern. To reduce drag, both propellers are enclosed in Kort nozzles and are independently steered with servos inside the hull.

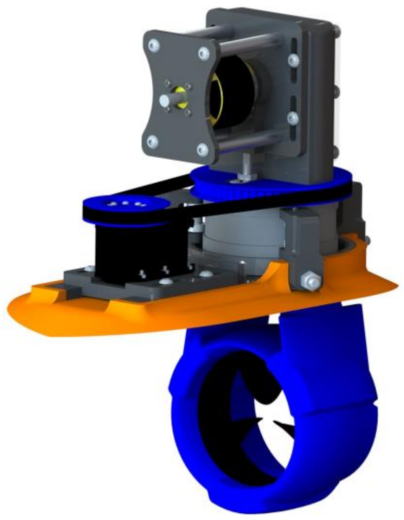

Figure 2. CAD model of the azimuth thruster used on PropaGator 2

To allow the boat to "sense" its environment, a suite of sensors were installed. A Spartan AHRS-8 IMU paired (through a Kalman filter) with a GPS module allowed PropaGator 2 to determine its position, heading, roll, pitch, and yaw. For object detection and recognition, an IDS imaging camera and a SICK LIDAR were installed onto the front of the boat. For communication between the shore and the boat, a wireless router was used.

Two types of drive systems were considered: a differential drive and a constrained azimuth drive system. Originally, a differential drive system was considered. The thrusters would have been locked in a forward facing direction during this configuration. Turning was handled by differential thrust. While control of the vessel would be simple, station holding and strafing would not have been possible with this type of drive system. Because of these limitations, the azimuth system was chosen, resulting in better maneuverability.

The azimuth thruster drive system took advantage of the thrusters' ability to create thrust vectors in different directions. This allowed the boat to maintain its heading z (rotation) while traversing in the x (surge) and y (sway) axis as shown in figure 3. However, because the thruster system is over-actuated, the solutions to produce the desired thrust vectors were not unique. Additionally, the thrusters could not rotate more than several full revolutions due to cable binding. To prevent binding, the thrusters were limited to 180 degrees of rotation at a rotation speed of 340 degrees per second. With the new constraint on the thruster rotations, producing thrust vectors became much more challenging. A method was needed that could determine appropriate vectors while using a constrained and over-actuated system. To achieve this goal, a constrained nonlinear controller was implemented on PropaGator 2.

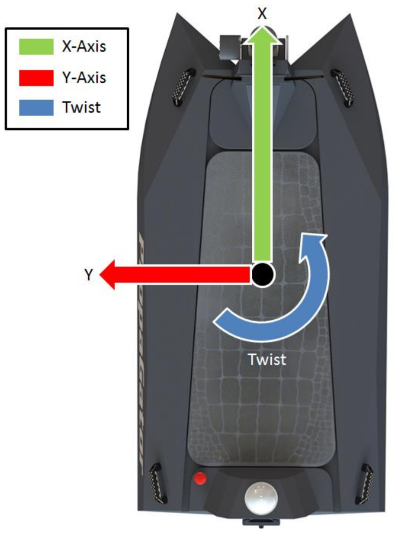

Figure 3. Three axes of motion performed by PropaGator 2

3. NONLINEAR CONTROL ALLOCATION

The constrained nonlinear control allocator, used on PropaGator 2, was created specifically for surface vessels with multiple thrusters and meant to be solved in real time [9]. Additionally, the allocator had a term which penalized solutions containing singularities. In order to use the allocator, the propulsion system had to be defined first. Each thruster had to be characterized by the maximum and minimum amount of thrust u, maximum and minimum angle limitations α, and the maximum and minimum rotation speed $\Delta\alpha$. The allocator initially developed for surface vessels is:

$$J(\alpha,u,s) = \sum_{i=1}^{m} W_i(u_i) + s^T Q_s + (\alpha - \alpha_0)^T\, \Omega(\alpha - \alpha_0) + \left(\frac{\varrho}{\varepsilon + det\left(B(\alpha)B^T(\alpha)\right)}\right), \qquad (1)$$

where

$$s = \tau - B(\alpha)u, \qquad (2)$$
$$u_{min} \le u \le u_{max}, \qquad (3)$$
$$\alpha_{min} \le \alpha \le \alpha_{max}, \qquad (4)$$
$$\Delta\alpha_{min} \le \alpha - \alpha_0 \le \Delta\alpha_{max}, \qquad (5)$$

The first term in equation (1) is the total power consumption, $W_i(u_i)$ of the actuators. The next term, $s^T Q_s$, adds a penalty based on the force difference s. s is found by subtracting the achieved force $B(\alpha)u$ from the desired force s. The weights in the matrix Q are chosen so that $s \approx 0$ whenever possible. In an effort to limit drastic changes, the third term penalizes changes in thruster azimuth angle. The matrix Ω is tuned until drastic changes are acceptably limited. The fourth term penalizes singularities, thereby affecting the maneuverability of the vessel through the use of the ϱ variable. A high ϱ would result in high maneuverability at a cost of more power. Whereas a low ϱ value would result in a less responsive but more efficient solution.

The above equation is nonconvex and solving it is generally intractable for real-time applications. The optimization is considered nonconvex because of the potentially nonlinear power cost in (1), the nonlinear constraint (2), and the singularity avoidance nonconvex term in (1). The high computation time caused by the nonconvex equation would make the optimization problem not useful to applications that required the solutions in rapid succession like PropaGator 2. In order to make the computation tractable, a linear approximation of the constraints and dynamics yielded a tractable quadratic program.

The resulting approximation is as follows [9]:

$$J_{QP}(\Delta\alpha,\Delta u,s) = \sum_{i=1}^{m}\left(\frac{dW_i}{du_i}(u_0,i)\Delta u_i + \frac{d^2W_i}{du_i^2}(u_0,i)\Delta u_i^2\right) + s^T Q_s + \Delta\alpha^T \Omega \Delta\alpha + \frac{d}{d\alpha}\left(\frac{\varrho}{\varepsilon + det\left(B(\alpha)\right)}\right)_{\alpha=\alpha_0} \Delta\alpha \qquad (6)$$

linear constraints

$$s + B(\alpha_0)\Delta u + \frac{\partial}{\partial\alpha}(B(\alpha)u)\Big|_{\substack{\alpha=\alpha_0\\ u=u_0}} \cdot \Delta\alpha = \tau - B(\alpha_0)u_0, \qquad (7)$$
$$u_{min} - u_0 \le \Delta u \le u_{max} - u_0, \qquad (8)$$
$$\alpha_{min} - \alpha_0 \le \Delta\alpha \le \alpha_{max} - \alpha_0, \qquad (9)$$
$$\Delta\alpha_{min} \le \Delta\alpha \le \Delta\alpha_{max}, \qquad (10)$$

4. IMPLEMENTATION

To make PropaGator 2 controllable on the water, three levels of software were required: the path planner, the controller, and Azi-drive. At the highest level, the path planner was used. The path planner constantly checked the position and orientation of the boat. Once given a new goal position from the user, the path planner produced a desired position and orientation (DPOSE) to guide the boat towards the goal position. The DPOSE was placed two meters away from the boat, between the current position and the goal position. As the boat moved towards the DPOSE, the DPOSE was moved closer to the goal position. When DPOSE was less than two meters from the goal position, the distance between the DPOSE and the boat was decreased at a linear rate determined by the path planner, until the boat was within a one meter radius of the goal position.

Moving the boat in the right direction based upon the position of the DPOSE was performed by the controller. Using a proportional derivative (PD) controller, the responsiveness of the boat could be changed. Differences in headings and position from the DPOSE were translated into a wrench (i.e., a force and torque vector). The wrench was then passed to the Azi-drive shown in figure 4.

Azi-drive converted the wrench values into thruster positions and thrust amounts by solving the associated quadratic program. In order to produce a thrust vector that best fits the given wrench, Azi-drive calculates the best value and translates that value into thruster forces. The best fit solution is based upon minimizing the error in the cost function. The singularity avoidance gain, ϱ, governs the cost for producing a solution in a singularity state, i.e., one where enormous (potentially infinite) thrust is required to produce particular wrenches.

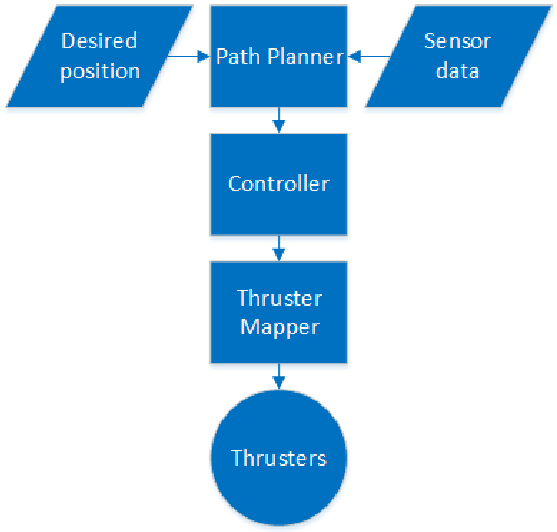

Figure 4. Flow chart of maneuverability, starting at the path planner and ending at thruster commands

Avoiding singularities was important for PropaGator 2 because singularities caused the boat to become unstable. A singularity with the thrusters was encountered when the thrusters were turned toward each other. The thrusters were still generating individual forces, but the boat did not move because of the orientations. Gradually, the controller increased the amount of thrust generated by each thruster in an attempt to move the boat. The increase in thrust did not move the boat and the cycle repeated, forcing the user to step in and stop the program (i.e., causing the autonomy to fail).

The thruster mapper solution rate was too slow when Azi-drive was first programmed. Python was used to initially solve the optimization problem. Python is a convenient computer to quickly develop programs. However, because Python is a runtime language, programs written in the language run significantly slower than compiled languages. Using Python, the computer on PropaGator 2 was able to solve the optimization problem at a rate of 10 Hz. The slow update rate caused oscillations in orientations of the boat that could not be avoided. A faster solution for the optimization problem was necessary.

The code-generation tool, CVXGEN, provided a way to rapidly solve the nonlinear control allocation equation. To use CVXGEN, a convex optimization problem is defined in a high level form. The high-level definition is passed to the CVXGEN tool, which in turn produces a quadratic program solver in C [10]. CVXGEN quickly solves convex optimization problems by unrolling matrix multiplications, directly accounting for sparsity, and taking advantage of problem structure at generation time. The CVXGEN-generated solver found solutions up to 25x faster than the previous Python implementation, allowing the solver to avoid numerical issues as a consequence of linearization. Once applied to the boat, the responsiveness increased tenfold.

Prior to testing PropaGator 2 in the water, changes applied were tested in a simulator (see Figure 5). While the simulator had basic physics capabilities built in, it did not give a perfect representation of how the boat would perform. However, the simulator allowed for immediate feedback to changes made in software. The simulator also provided a chance to practice tuning the PD controller gains prior to in-water testing.

After testing the boat in simulation, PropaGator 2 was transported to a local pool for testing. The pool provided a controlled environment where gain changes to the PD controller were more apparent due to the placidity of the water. Another advantage of using the pool was that students could be in the water with the boat to physically disturb the vessel in order to observe its reactions to changes in heading and position.

Because of the thruster configuration, tuning one of the axes affected the other axes. This required all three axes to be tuned simultaneously. The processing of tuning all three axis continued until the boat was acceptably able to maintain its position and heading.

Once the team was satisfied with the performance of PropaGator 2 at the pool, the vessel was placed in a small lake. The lake provided a large enough area to allow the boat to travel long distances across the water. During the longer movements, the team was able to observe how PropaGator 2 tracked across a straight line. If the boat tracked poorly, the gains in the PD controller were slightly adjusted accordingly until the performance of the boat was satisfactory.

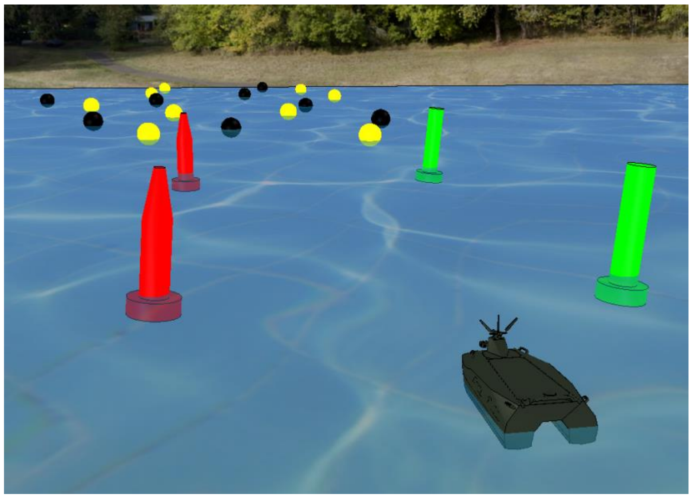
Figure 5. Simulator used to test software prior to boat testing

5. RESULTS

In order to prove that Azi-drive was working properly, the actual wrenches were compared to the desired wrenches. To find the actual wrench values produced by Azi-drive, the following equations were used:

$$\tau_{net} = \tau_{stbd} + \tau_{port}, \qquad (11)$$
$$\tau = F_{tx} r_y + F_{ty} r_x, \qquad (12)$$
$$F_{xnet} = F_{stbdtx} + F_{porttx}, \qquad (13)$$
$$F_{ynet} = F_{stbdty} + F_{portty}, \qquad (14)$$
$$F_{tx} = -(u)\cos\theta, \qquad (15)$$
$$F_{ty} = -(u)\sin\theta. \qquad (16)$$

Equation (11) is the actual torque experienced by the boat, τ_{net}. This is found by taking the sum of the torque created by both thrusters. The torque created by either the port or starboard thruster was found by multiplying the force in the x-direction, F_{tx}, with the thruster's y-distance from the center of rotation, r_y. Torque from the y-direction force, F_{ty}, is found by multiplying the force by the x-distance, r_x. Summing the two products produced the thrusters torque as shown in (12). In equations (13) and (14), the sum of the forces along the x axis and y axis are shown. Finally, in order to find the force in each direction, equations (15) and (16) were.

Data from testing was recorded for post analysis of the Azi-drive. The topics recorded were: desired wrench, thruster force, and thruster direction. Since the rates of recording for each movement was different and inconsistent, a script was created. The script was created to find the actual wrench in real time by subscribing to the topics that produced the thruster force and thruster angle. In the following paragraphs, the results of the script are discussed.

To show that Azi-drive was working, the actual forces and torque were compared to the desired forces and torque. To make the graphs easier easier to read, the actual forces and torque were averaged with the previous five samples, shown in Figures 6, 7, and 8, where the y-axis is the force or torque and the x-axis is time:

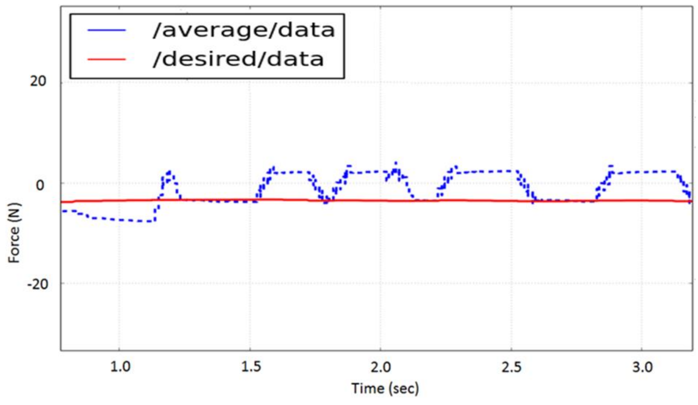

Figure 6. The solid line is the desired X force and the dashed line is the averaged actual X force

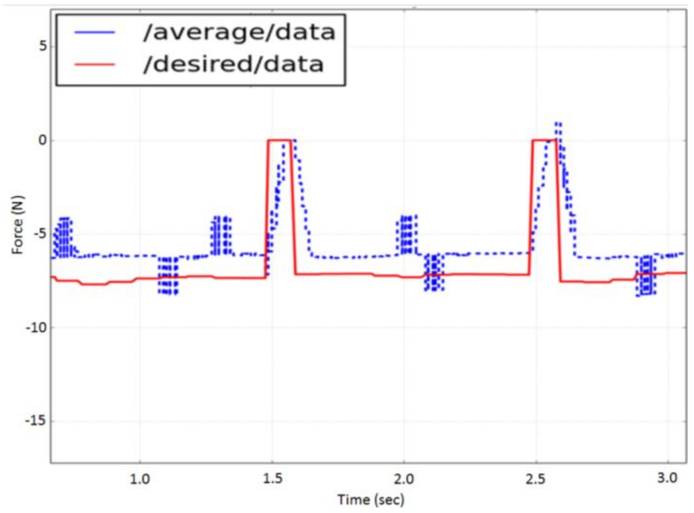

Figure 7. The solid line is the desired Y force and the dashed line is the averaged actual Y force

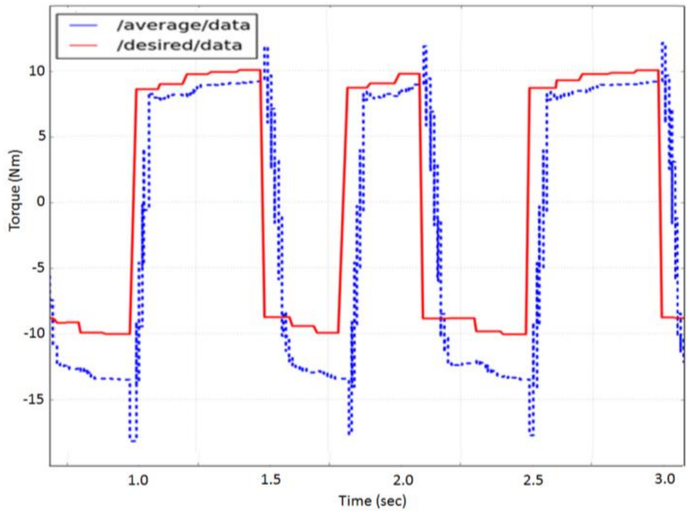

Figure 8. The solid line is the desired torque and the dashed line is the averaged actual torque

As the graphs above show, Azi-drive was able to produce forces and torques that were close to the desired results given by the controller. The actual X-force varied slightly from the desired X-force. This variation can be attributed to the optimized solution compensating with the X-force in order to maintain a closer relation with the Y-force and torque in Figures 6 and 7. In Figure 7, the Y-forces created can be clearly seen to match the desired forces. A small amount of latency can be seen as the thruster mapper responds to wrench changes from the controller. Finally, the actual torque is shown to be very similar to the desired torque. Once again, latency is seen as the Azi-drive attempts to match the desired torque request from the controller. The above graphs prove that Azi-drive works well as a thruster mapper and has potential for use on other platforms.

6. CONCLUSION

This paper covered the research, implementation, and testing of a novel thruster mapper used to control the ASV, PropaGator 2. Two azimuth thrusters were selected as the propulsion system because of the maneuverability gained when compared to a traditional differential drive system. A constrained non-linear control allocation optimizer was implemented in order to control the thrusters while constraining them to a limited amount of rotation. Once Azi-drive was tested and applied to the boat, PropaGator 2 was able to move in all three dimensions (surge, sway, and yaw) when traversing from point to point.

Having only two thrusters in the back of the boat made controlling of PropaGator 2 difficult. Adding one or more thrusters to the bow would simplify the control problem. Even if the thrusters were fixed and could only produce a force along one axis, this force would assist with swaying the boat, a movement that proved difficult in the current configuration. Having a dynamic two stage control system would help to overcome the issue with the boat handling differently at higher speeds due to planing. At higher speeds, the boat generally does not need to strafe along the y-axis and could be maneuvered using a differential drive behavior. A control method that allowed movement along the y-axis would be activated when operating at lower speeds. Finally, improving the path planner by allowing it to factor in the velocity of the boat would make transitioning between points smoother. Currently, the path planner does not account for the momentum of the boat. As the boat approaches the waypoint, it tends to overshoot the point due to the boat inertia.

7. ACKNOWLEDGEMENTS

The authors would like to thank the students and faculty members who helped with the operation of the boat, the University of Florida College of Electrical and Computer Engineering and the College of Mechanical and Aerospace Engineering departments. An extended thank you to Dr. Schwartz and Dr. Arroyo and to the Machine Intelligence Lab. Thank you Dr. Crane for transportation and the use of the lab equipment. A special thank you to Jacob Mattingley for generously providing an academic license for CVXGEN.

REFERENCES

[1] Berge, S. P., & Fossen, T. I. (1997). Robust control allocation of overactuated ships; experiments with a model ship. In Preprints of IFAC conference on maneuvering and control of marine craft, Brijuni, Croatia.

[2] Gray, A., Shahrestani, N., Frank, D., and Schwartz, E., 2013, “PropaGator 2013: UF Autonomous Surface Vehicle,” AUVSI Foundation’s 6th Annual RoboBoat Competition, Virginia Beach, VA.

[3] Walters, P., Sauder, N., Nezvadovitz, J., Voight, F., Gray, A., and Schwartz, E., 2014, “SubjuGator 2014: Design and Implementation of a Modular, Fault tolerant AUV,” AUVSI Foundation’s 17th Annual RoboSub Competition, San Diego, CA.

[4]: Frank, D., Gray, A., and Schwartz, E., 2014, “PropaGator 2014: UF Autonomous Surface Vehicle,” AUVSI Foundation’s 7th Annual RoboBoat Competition, Virginia Beach, VA.

[5] Nezvadovitz, J., Griessler, M., Voight, F., Walters, P., and Schwartz, E., 2015, “SubjuGator 2015: Design and Implementation of a Modular, High-Performance AUV,” AUVSI Foundation’s 18th Annual RoboSub Competition, San Diego, CA.

[6] Frank, D., Gray, A., and Schwartz, E., 2014, “PropaGator 2: A Planing Autonomous Surface Vehicle With Azimuth Rim-Driven Thrusters,” Proc. 14th Annual Early Career Technical Conference, Birmingham, Alabama, pp. 185-190.

[7] Patel, D., Frank, D., & Crane, C. (2014, October). Controlling an over-actuated vehicle with application to an autonomous surface vehicle utilizing azimuth thrusters. In *Control, Automation and Systems (ICCAS), 2014 14th International Conference on* (pp. 745-750). IEEE.

[8] Lamb, T., 2004, Ship Design and Construction Vol. II, Society of Naval Architects and Marine Engineers, Chap. 45.

[9] Johansen, T., Fossen, T., & Berge, S. P. (2004). Constrained nonlinear control allocation with singularity avoidance using sequential quadratic programming.*Control Systems Technology, IEEE Transactions on*, *12*(1), pp. 211-216.

[10] Mattingley, J., & Boyd, S. (2012). CVXGEN: a code generator for embedded convex optimization. *Optimization and Engineering*, *13*(1), pp. 1-27.