ASEE 2014 Zone I Conference, April 3-5, 2014, University of Bridgeport, Bridgpeort, CT, USA.

Electrical Engineering Capstone Project on Dynamic Position System

Christopher Palmieri, Judith Hooymans, Cole Gingrich, Tooran Emami, Aaron Dahlen, and Joseph Staier
Department of Engineering- Electrical Engineering Section
United States Coast Guard Academy
New London, CT, USA
Christopher.A.Palmieri@uscga.edu
Judith.A.Hooymans@uscga.edu
Cole.R.Gingrich@uscga.edu
Tooran.Emami@uscga.edu
Aaron.P.Dahlen@uscga.edu
Joseph.E.Staier@uscga.edu

Abstract—**This paper presents the milestones of a senior capstone project in Electrical Engineering at the United State Coast Guard Academy. Three students worked on their senior capstone project for two semesters. Students applied a combination of knowledge they learned in their undergraduate education to construct an autonomous dynamic positioning vessel. This project consists of the construction and networking designs. The construction designs of this vessel are exterior and interior parts. The exterior part of vessel was constructed with a 20 gallon salvage drum, PVC pipe frame, six 750 GPH motor cartridges, and two sensors, in order to control the heading and position of the craft in the water. The interior part of the vessel includes micro-controllers connected to the MD10C H-bridges, to control thruster speed and are also connected to an on board computer. All inside components are attached to a fuse block in order to protect the components. The networking design consists of various programs to handle the data for the vessel and how the various systems communicate with each other. The controller is implemented in MATLAB, data is stored in MySQL, a python script directs the data, and the Arduino micro-controller transfers data to the motors and from the sensors. A controller has been designed that successfully instructed the vessel to stay parallel to the walls of the tank water laboratory test. This controller only uses the heading motors and leaves the positional motors off.**

I. INTRODUCTION

The United States Coast Guard Marine Safety Center is the sponsor of an electrical engineering senior design project to create a vessel capable of autonomously maintaining heading and position using Dynamic Positioning System (DPS). Currently, the U. S. Coast Guard uses DPS to maintain navigation aids by holding the ship's position while the crew services and redeploys the navigation aid.

Additionally, the U. S. Coast Guard is required to inspect large drilling platforms at sea for their adherence to safety regulations and certify these platforms. There is a vested interest in the prevention mission of the Coast Guard to increase knowledge of how these large rigs position, so the inspectors can understand the different systems and how they integrate together.

Offshore oil drilling began in the early 1960's when at sea drilling began using anchored vessels. The need for DPS arose from the expansion of deep sea oil drilling. The platforms were starting to drill in places where it was impossible, due to the depth, to anchor. Initially, vessels were designed with four thrusters that were used to keep the platform in place. Instead of using Global Positioning System (GPS), the first DPS system had a ring of buoys around the outside and the system measured ranges to these buoys. The thrusters were capable of rotating a full 360 degrees so that they could maintain a certain range to each buoy. Although this system could keep the platforms in place, the thrusters were controlled manually [1].

Designs soon began on an automatic system that could hold the vessel in place without the manual control. To do this, the different thrusters would need to control the surge, sway and yaw of the vessel. With successful design and implementation, DPS became prevalent in the deep water drilling industry. It has been estimated that there are more than 2,000 platforms equipped with DPS in operation today[1].

There are variety types of controllers that are used in the DPS in large oil drilling rigs. The most common controllers used in the DPS systems are proportional integrator derivative (PID) controllers designed with a low pass or notch filter [2]. Other controllers researched include the Discrete-Time Variable Structure Controller, which reacted well to changing reference points, and a modified fuzzy gain scheduling PID controller, which was more responsive than the traditional PID controller [3].

This paper presents the milestones of a senior capstone project in Electrical Engineering at the United State Coast Guard Academy. Three students are working on their senior capstone project for two semesters. Students applied a combination of some of the courses they have learned in their undergraduate education to construct an autonomous dynamic position vessel. The milestones for this project were as follows. 1) Design and build the interior and exterior of the vessel. 2) Design a network that operates the vessel by a given user input. 3) Test the vessel in an indoor tank water laboratory by remote controller. 4) Obtain the dynamic of the platform by system identifications. 5) Implement the final controller to maintain with a GPS location on an open water river. The first three milestones were accomplished in fall 2013. The last two milestones would be accomplished in spring 2014.

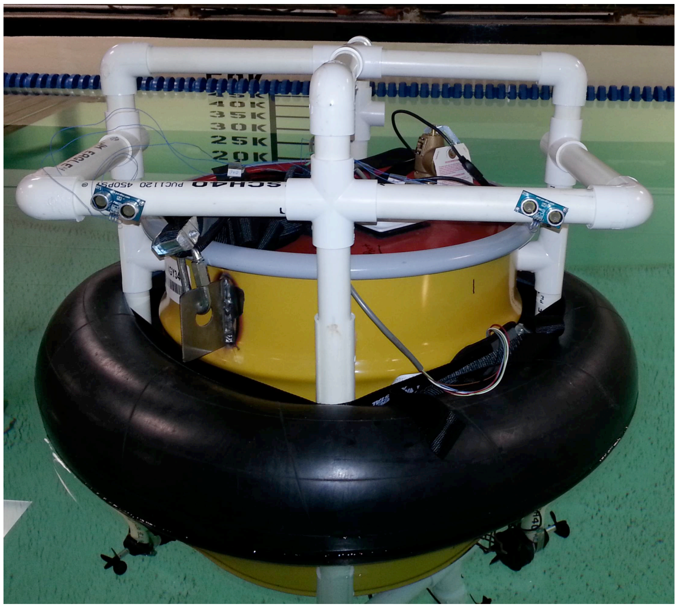
This paper is organized as follows. The functional designs that are included the construction and networking designs are presented in Section II. Indoor Laboratory results are presented in Section III. Future plan is summarized in Section IV. The conclusion of this paper is presented in Section V.

II. FUCNTIONAL DESIGNS

A. Construction Design

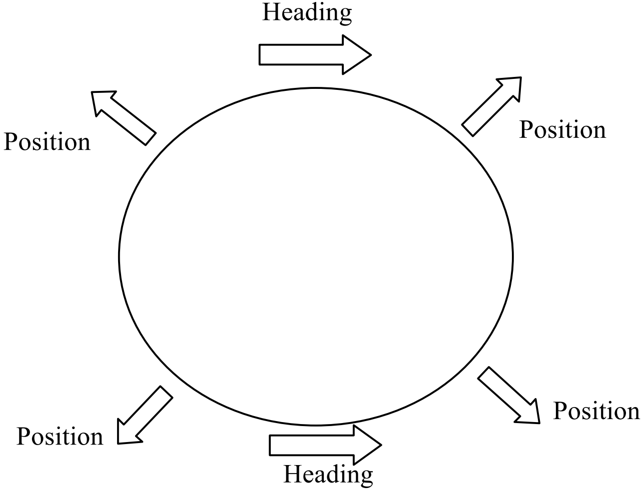
The first task is to actually build the craft that autonomously maintains position and heading set by the user. The first step is to build the exterior of the craft. The current design of the vessel shows in Figure 1.

Fig. 1. Dynamic position vessel



The most important building point of the exterior is how large the craft would be after construction. A small enough vessel was chosen to allow easy transportation from the design laboratory to the various testing sites. Ultimately, a 20-gallon salvage drum was chosen to house all of the interior components because of its portability, durability and watertight properties. PVC pipe frames were built to house the propellers and sensors on the outside of the salvage drum. The PVC pipe frame enclosed the barrel and allowed the thrusters to be attached to the craft without having to drill into the side of the barrel, compromising watertight integrity. An inner tube was placed around the frame to hold it in place and provide much needed stability. Six GPH motor Cartridge thrusters were mounted on the PVC pipe frame, two for the heading control and 4 thrusters for the position control. The modeling of these motors is shown in Figure 2. Two ultrasonic sensors were mounted on the PVC pipe frame; these sensors will be replaced with the New Mountain 100 Weather Station sensor in the final milestone design.

Fig. 2. Modeling of the 6 motors for heading and position controls



The inside of the craft was constructed in order to house an autonomous control equipment of the vessel. The inner shelves structure was made of hardboard separated by PVC pipe dividers. The shelves are held in place by thin tubes of solid plastic, which are fed through two holes that were drilled on opposite corners of each shelf. A hardboard frame sits on the bottom of the barrel to confine the batteries to their center positing. This hardboard frame is reinforced in two places with wooden blocks. The inside components are powered by two twelve volt batteries, sitting at the bottom of the structure to provide stability. The automotive computer performing all the calculations and data storage for the autonomous vessel lives on the second shelf just below the fuse block. All inside components are attached to the fuse block in order to protect the components.

The two Arduino micro-controllers in Figure 3 are connected to the six MD10C H-bridges, to control thruster speed and are also connected to the computer. There is one H-bridge for each motor; one of the H-bridges is shown in Figure 4. The H-bridges are fed commands from the Arduino and change the current to the thrusters to provide different levels of power for the motor. The inside of the vessel is shown in the Figure 5.

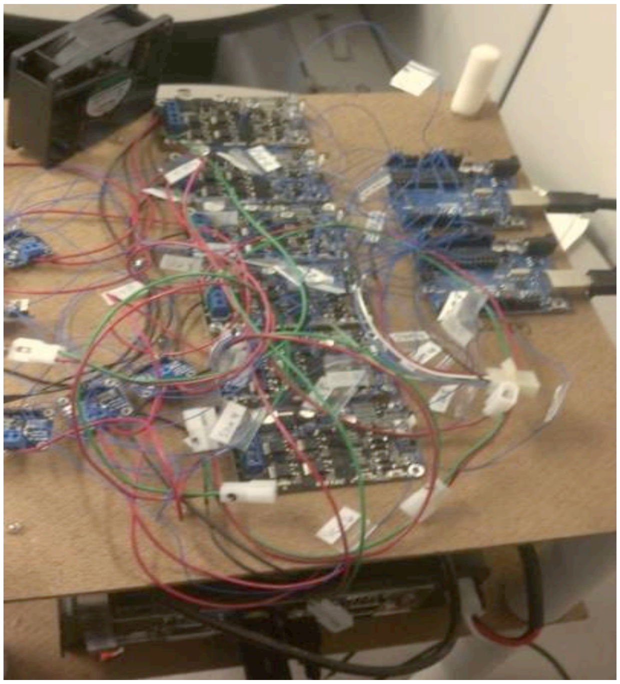


Fig. 3. H-Bridge to Arduino Interface

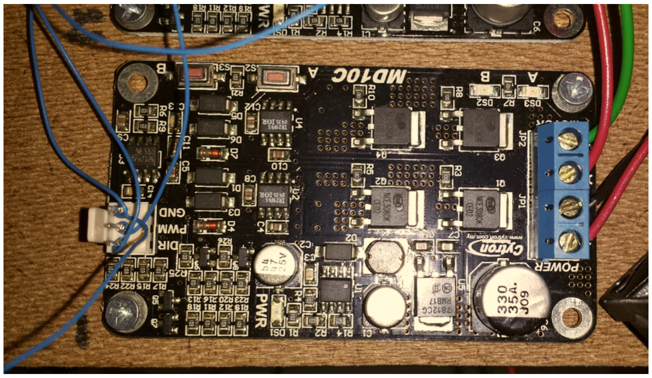


Fig. 4. MD10C H-bridge to control thruster speed

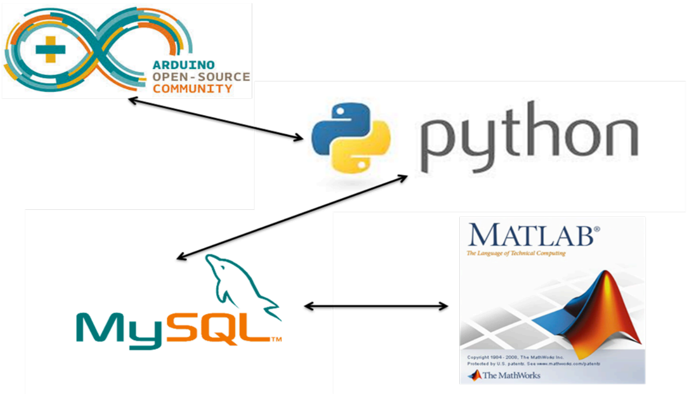

Fig. 5. The inside of the vessel

B. *Networking Design*

To begin creating the autonomous vessel, an on board computer is needed to do the calculations necessary for the controller that will later be implemented. Students decided upon a minimalistic version of CentOS 6.4 without a graphical user interface in order to maximize the speed of our computer's calculations. This computer is needed to run MATLAB in order to perform the calculations necessary for creating a controller. Additionally, a version of MySQL relational database is chosen as the storage of data; data invaluable to the refining of the controller. Python programming language allows successful communicating between MySQL and the Arduinos connected through the two COM ports. Using a Linksys router allows setup of a network named *"NETGEAR91"* for communication between the craft and a user's laptop. Students altered the settings of the command line-based CentOS, so that upon powering on it would automatically sign in and connect to the network that has been created. This enables the user to Secure Shell (SSH) into the automotive computer and starts the autonomous control system.

The networking of the vessel communication shows in Figure 6. This simple diagram of the various programs handles the data for the craft and how the various systems communicate with each other. The controller is implemented in MATLAB, data is stored in the MySQL as data base, a Python script directs the data, and Arduino transfers data to the motors and from the ping sensors.

Fig. 6. Networking model of autonomous vessel



For the parallel ping test, the running Python script checks the control value store in the MySQL database to determine if the craft is set to run in autonomous, manual, or all stop mode. If the craft is running in autonomous mode, the Python script sends the command "ping" to the Arduino using a COM port. Once Arduino receives the "ping" command, Arduino fires both the left and right ping sensors at the wall and sends the two values back to Python through the COM port.

Once Python receives the data from the ping sensors, it writes the results into a MySQL database using the *pysql* library. Python also checks a separate MySQL table within the database to determine the last motor command written to the database. Python then extracts the last command and passes it through the COM port to the Arduino with the leading tag "motor". Another database exists in MySQL and holds manual motor commands. When the control value reflects manual entries, the Python script checks the manual motor commands database and sends those commands to the Arduino in the same fashion. Arduino communicates solely with the Python over the COM port and with the H-bridges controlling the six motors. Python talks with Arduino using the COM port and to MySQL using the installed library.

Additionally, if the Python script detects that the control value has been set to all stop, the script will send Arduino a stop command which tells Arduino to stop all motors. Changing the control value or the manual motor commands is accomplished by updating the internal MySQL database located on the automotive computer inside of the craft.

The actual controller is implemented in MATLAB due to its adept ability to handle sine and cosine waves in addition to fast calculations. MATLAB communicates with the MySQL database and reads in the newest ping responses, averages the left and right ping values and calculates which way the craft needs to rotate to remain parallel to the desired wall. After the calculations, MATLAB calculates which command to send to the motors and writes the command into the MySQL database for autonomous motor commands. MATLAB will continue to function even when the control value is set to manual or all stop. However, the Python script will not read from the directory that MATLAB is writing to, which allows the other protocols to function without interfering with the controller.

Since both MATLAB and the Python script are both reading to or writing from the database, students searched for a data storage center capable of handling two different pieces of software and interacting with them simultaneously. MySQL is chosen as the database storage center. The MySQL database contains four tables. One table contains either the letter *'a'* for autonomous running, *'m'* for manual commands, and *'a'* for all stop in cases of an emergency.

One of the reasons MySQL was chosen is its ability to allow multiple programs to both read and write from the database simultaneously, necessity for the controller implementation. MATLAB and the Python script both need to simultaneously talk to MySQL without having to worry about coordination. Additionally, the recording of the differences in the ping sensors and the various motor commands makes it possible to allow for quick real time analysis using MATLAB in order to aid the controller.

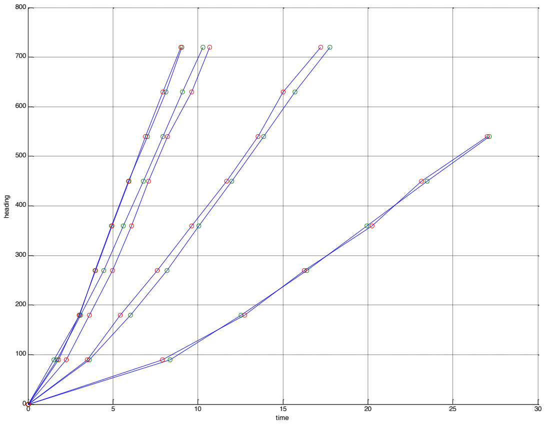
The database also contains a table to hold the ping sensor data. Those data entries are stored as left and right ping values. The other two tables are the manual and automatic motor command tables. The reason for having to separate tables is that it facilitates switching from manual to automatic control without having to shut down the automatic controller. When the control value described above is switched to the manual position from automatic, the Python script immediately begins to read from the manual commands table while MATLAB continues to write commands to the unread automatic commands table.

III. INDOOR LABORATORY RESULTS

To understanding how the vessel responds, students need to obtain both an open loop and a closed loop response for the heading motors. An open loop response is obtained by implementing a remote control interface, commanding a step, and measuring the plant's response. The open loop response is characterized by obtaining the time elapsed between each 90 degrees of rotation until the craft achieved its maximum linear angular velocity. Multiple tests were performed to capture the craft's power speed curves.

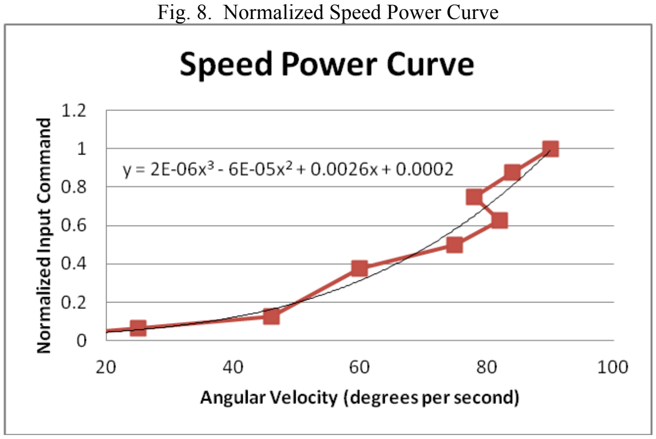
Students obtain velocity by taking the derivative of angular position shown in Figure 7. This position graph demonstrates that the craft quickly reaches a constant maximum angular velocity for each particular motor setting. Note that angular velocity is the slope of the angular position line. In a linear system the angular velocity would be a linear function of the plant's input command. This is to say, the craft would spin at half its maximum angular velocity when commanded to half power. This is not the case as shown in Figure 7. The heading position lines appear concentrated at around full power, thus students conclude that the system is in fact non-linear.

Fig. 7. Heading Position for Full, Half, Quarter, and Eighth Power



This non-linearity is demonstrated in Figure 8 which presents the speed power curve derived from the data shown in Figure 7. Here the maximum angular velocities for the various motor power levels are shown. The next step for the group will be to find the inverse transfer function of the speed power curve in order to linearize the plant's heading angular velocity.

Fig. 8. Normalized Speed Power Curve



The input commands for the motor have been normalized and graphed to show the maximum angular velocity for each input motor command. As mentioned previously this system is non-linear as evidence in Figure 7. The curve fit to the data shows a non-linear response to the motor commands. The next step for the group will be to find the inverse transfer function of the speed power curve in order to allow us to input the correct motor commands for this non-linear system.

Currently, a controller has been designed that successfully instructs the craft to stay parallel to the walls of the tank water laboratory test. This controller only uses the heading motors and leaves the positional motors off. Using the open loop response they will obtain an open loop transfer function for our non-linear system. Finding the transfer functions for the plant will enable us to properly create a controller. After students create a controller that moves from its actual location to its desired position, they will change the input data to the system from ping distances to magnetic heading and GPS location.

IV. FUTURE PLAN

A rudimentary closed loop heading controller has been constructed and successfully tested. Using two ping sensors for positional feedback, the craft has the ability to maintain a parallel orientation with the walls if the pool. In future implementations control of the translational motors will be implemented allowing the craft to maintain both a parallel orientation and a specified distance. These rudimentary controllers are useful first steps as they allow students to capture data with improved resolution. This allows further refinement of the plant model and improving the initial estimates based on the elapsed time angular measurements. When the pool tests are complete the ping sensors will be replaced magnetic heading and GPS sensors and open water tests will be conducted. This is expected to be a trivial change as students view the ping sensor data as a vector (heading angle and distance) and is equivalent to the GPS information. The craft will have a vector indicating distance and angle to desired position.

The disturbances from wind and current have been negligible in the laboratory test. After the vessel has a controller working in the pool and sheltered outside locations, students will continue to redefine the controller until it runs autonomously and corrects for disturbances caused by wind and water currents.

V. CONCLUSIONS

This paper presents three of five milestones of a senior capstone project in Electrical Engineering at the United State Coast Guard Academy. Three students applied a combination of most of courses they have learned in their undergraduate education to construct an autonomous dynamic position vessel.

This is an excellent capstone project as students have applied the lessons learned from nearly all of the courses in their undergraduate education. From computer science classes they have demonstrate the ability to construct and design networks as well as program in a variety of languages including MATLAB, Python, and C. A MySQL database solution is also included allowing easy post processing of data. From electronics they demonstrate the ability to interface sensors and actuators via a microcontroller interface. From signal processing / control system track they demonstrate the ability to control a non-linear craft.

ACKNOWLEDGEMENT

The authors extend special thanks to all U.S. Coast Guard Academy employees who helped our team with building and testing the vessel.

REFERENCES

[1] Shatto, Howard, “History of Dynamic Positioning,” Dynamic Positioning Committee of the MTS, 2011. http://www.dynamic-positioning.com/history.html

[2] Åström, Karl Johan, and Tore Hägglund, "The future of PID control." *Control engineering practice,* VOL. 9, NO. 11, pp. 1163-1175, 2001.

[3] Tao Yung, “Architectures of computational verb controllers: Towards a new paradigm of intelligent control,” *International Journal of Computational Cognition,* VOL. 3, NO. 2, June 2005.