ASEE'S VIRTUAL CONFERENCE

At Home with Engineering Education

JUNE 22 - 26, 2020 #ASEEVC

Paper ID #31129

STRIDER(Semi-autonomous Tracking Robot with Instrumentation for Data-acquisition and Environmental Research)-Pitfalls and Successes of a Vertically Integrated Experiential Learning Project spanning Multiple Years

Mr. Brandon Miles Gardner, University of Maryland Eastern Shore

Junior general engineering student involved with undergraduate research

Dr. Abhijit Nagchaudhuri, University of Maryland, Eastern Shore

Dr. Abhijit Nagchaudhuri is currently a Professor in the Department of Engineering and Aviation Sciences at University of Maryland Eastern Shore. He is a member American Society for Mechanical Engineers (ASME), American Society for Engineering Education (ASEE) and, American Society for Agricultural and Biological Engineers(ASABE) and is actively involved in teaching and research in the fields of (i) robotics and mechatronics, (ii)remote sensing and precision agriculture, and,(iii) biofuels and renewable energy. He has published more than 70 refereed articles in journals and conference proceedings. Dr. Nagchaudhuri received his baccalaureate degree from Jadavpur University in Kolkata, India with honors in Mechanical Engineering. Thereafter, he worked in a multinational industry for a little over three years before joining Tulane University as a graduate student in the fall of 1987. He received master's degree from Tulane University in 1989 and doctoral degree from Duke University 1992.

Mr. Jesu Raj Pandya, University of Maryland Eastern Shore

Currently a doctoral student in Food & Agriculture Sciences, University of Maryland Eastern Shore (UMES), MSc - Applied Computer Sciences at UMES, BSc - Electrical and Electronical Engineering (JNTU). Interest in robotics and automation in food production and food safety.

Mr. Rakesh Joshi, University of Maryland Eastern Shore

I am a lecturer at the University of Maryland Eastern Shore. My interests include Machine learning and Artificial intelligence. I also work as a consultant for a software development company in the financial Industry.

Mr. Fredrick Landon Bickle, University of Maryland Eastern Shore

Fredrick Landon Bickle currently resides in Annapolis, Maryland. In 2019, he graduated Anne Arundel Community College with an Associate's degree in Engineering transfer. He is now attending University of Maryland Eastern Shore with a major in Engineering with Aerospace specialty.

Dr. Mark E. Williams, Univ of MD Eastern Shore

After earning my PhD in Physics at the University of Cincinnati, I became more interested in using a computer for research and became a computational physicist. I have been at UMES for 20 years teaching Mathematics, programming languages and some physics. I collaborated with researchers at Univ of Alabama in spintronics and am currently doing interdisciplinary research with undergraduates and the USDA at UMES.

STRIDER (Semi-autonomous Tracking Robot with Instrumentation for Dataacquisition and Environmental Research)-Pitfalls and Successes of a Vertically Integrated Experiential Learning Project spanning Multiple Years

Abstract:

The Semi-autonomous Tracking Robot with Instrumentation for Data-acquisition and Environmental Research (STRIDER) is conceptualized as a wireless controlled aquatic robot with the capability of taking water samples as well as in-situ water quality data. STRIDER can be operated into either salt or freshwater to provide water quality data at the surface as well as at specified depths over a representative area over a chosen water body. The STRIDER team consists of a small group of engineering majors as well as students from other fields collaborating to meet the requirements set by scientists at the Environmental Monitoring and Food Safety Laboratory (EMFSL) of United States Department of Agriculture (USDA); under the advice of a few faculty members at the University of Maryland Eastern Shore (UMES). STRIDER currently has the capability of providing critical geo-located measurements; pH, Oxidation Reduction Potential (ORP), and Dissolved Oxygen (DO) values at the surface and other specified depths. This data can be interpolated over the surface, as well as across the depth to provide a three-dimensional representation of the variation of water quality parameters of a chosen water body. This project has allowed for the development of multidisciplinary research and experiential learning framework to engage students at UMES within and outside the classroom. Many different components of engineering design, environmental sciences, and other related fields are integral to this project. An overview of the design challenges and accomplishments of the team are provided in this work.

Introduction:

STRIDER project has been ongoing at UMES in collaboration with USDA EMFS Laboratory scientists and staff since 2014 and has provided a platform for experiential learning, research, and engineering design with the overarching goal of semi-autonomous sampling and water quality measurements in relevant water bodies [1, 2]. The design efforts have undergone several iterations for improved performance and capabilities. As students involved and leading the project have moved on and graduated new student groups have filled in under the advice of involved UMES faculty and USDA scientists. The newest version of STRIDER that is under development version utilizes a commercial-grade water quality sampling unit called an ISCO 6712 Full-size Portable Sampler. This paper outlines and illustrates the recent efforts with the STRIDER project primarily devoted to improving the existing platform [3,4] under the leadership of the primary author.



Figure 1: Field Trial with STRIDER

Objectives:

Objectives for STRIDER's mission are to provide a cost-effective method for semi-autonomous water quality sampling and data collection on an in-situ and ex-situ basis. Utilizing STRIDER to obtain geo-located measurements of Dissolved Oxygen(D.O.), Oxidation Reduction Potential(O.R.P.), and pH measurements at specified depths can be mapped and interpolated for three-dimensional analysis of the water body. STRIDER is also to be eventually delivered to the USDA researchers and staff to do a bacterial analysis of the water samples collected from the water bodies using the system.

Materials:

STRIDER's pontoons are constructed of sealed foam and positioned according to the distribution of weight on the platform. The platform is affixed to three metal square tube beams of aluminum welded to three cross braces. The platform base is the foundation that is attached to the tri-toons.



Figure 2: Student Team Moving STRIDER



Figure 3: Electronics box containing Central processing unit and electronic speed controllers

The winch is comprised of two sets of triangle frames and a central shaft that allows a drum to be rolled up and down. Cables for sensors and tubes for the water quality run through the center of the shaft such that the cables do not experience extreme loading when turning the winch or rotation at the joint. Water samples are pumped using a peristaltic pump mounted on the starboard side behind the winch. The turntable is comprised of a stepper motor and gearbox that connects to a circular board with mounts for bottles concentrically. Through the center is the output of the peristaltic pump which deposits water samples into rotating bottles. At the stern of the boat, there is a mount for a 12v glass-mat battery and the electronics box. On the port and starboard side of the stern are two 12v high torque dc motors which power the paddlewheels made of aluminum [see Figures 1 and 2].



Figure 4: Designed wire diagram for power distribution system.



Figure 5: Designed wire diagram for STRIDER's wire harness.

The speed controller and the electric set up and the wiring diagrams for the STRIDER set up is shown in Figures [3-5].

STRIDER's electronic box has a waterproof quick connect at the left-hand side of the electronics box shown in Figure [4]. The wiring for this quick connect is shown in Figure [6]. This is done to ensure that when the box is closed; under no uncertain circumstances will the electronics short due to water. There is also a fuse located in series with the power distribution system that will stop all power if current exceeds the fuse limit. This power distribution system is shown in Figure[5]. The power distribution system splits the signal coming from the battery into various reference voltage(5v), 12v and 6v lines which are routed to busses and used throughout in the various processors, speed controllers, fans, and other components.

Methods:

The efforts undertaken by the new student team to improve the STRIDER platform reported earlier [3] utilizes a tri-toon boat platform for added stability and buoyancy. A turntable with a fixed dispenser is used with 15 different possible water samples (see Figure 6). It may be noted the version reported earlier could carry only 7 bottles [3]. The platform can collect water samples from five locations at three different depths. This data can be interpolated for three-dimensional analysis of the water quality of the body of water in which the system is deployed. STRIDER utilizes a programmable logic device (PLD) with a bootloader called an Arduino Mega. This PLD has been modified through the use of platform IO to run an Arduino/Object-Oriented C variant which has been deemed Object-Oriented Arduino code compiled outside of the Arduino in a virtual bootloader in Visual Studio Code. This code is then injected into the Arduino on the armature that matches the specification of the virtual bootloader. The Mega can control both the navigational aspect of the boat as well as the scientific logging and acquisition through the use of the DO, ORP, and pH sensors. These in-situ measurements provide primary data sources that can be used quickly and efficiently for analysis in conjunction with the ex-situ samples that will be processed by the USDA EMFSL staff.



Figure 6: 15 Bottle turntable dispensary system.

Results:

A portion of the STRIDER code developed by the team is shown in Figure 7. The code activates the hardware and the sensor system for the sampling set up with the 15 bottles and ensures that for every individual sample the suction snout is directly over the respective bottles so that peristaltic pump when activated fills the appropriate bottle for a specified length of time. A stepper motor is

used with a gearing arrangement similar to a Lazy-Susan for appropriately indexing the bottles during the field operations. The boat can be operated both with remote control and autonomously to move it to 5 GPS locations where it takes samples from 3 different depths at each location. At the time of writing this paper the field trials for sample collections have been successfully accomplished for the improved platform. Modifications are underway to write the GPS locations for each of the 5 sampling locations covered for a single run and the DO, pH, and ORP values obtained from sensors deployed using the winch system at the various depths on an SD card. The data from the SD card will be processed and mapped using software such as GPS Visualizer and/or ESRI- ArcMap.

<pre>void loop() { lbotor.writeticroseconds(Trigger.Channel3(MicroSeconds)); rbotor.writeticroseconds(Trigger.Channel3(MicroSeconds)); rbotor.writeticroseconds(Trigger.Channel3(MicroSeconds)); //Ave the table to its initial home position for precision of table while (digitalRead(tableCheck) == HIGH)</pre>	175	}
<pre>177 Pwtor.writeticroseconds(Trigger.channel3(MicroSeconds)); 178 rwtor.writeticroseconds(Trigger.channel4(MicroSeconds)); 179 179 179 170 ///wre the table to its initial home position for precision of table 181 ///wre the table to its initial home position for precision of table 182 while (digitalRead(tablecheck) == HIGH) 183 { 183</pre>	176	void loop() {
<pre>rMotor.writeMicroseconds(Trigger.Channel4(MicroSeconds)); //Nove the table to its initial home position for precision of table while (digitalRead(tableCheck) == HIGH)</pre>	177	lMotor.writeMicroseconds(Trigger.Channel3(MicroSeconds));
<pre>//Move the table to its initial home position for precision of table while (digitalRead(tableCheck) == HIGH)</pre>	178	rMotor.writeMicroseconds(Trigger.Channel4(MicroSeconds));
<pre>//Wove the table to its initial home position for precision of table //Wove the table to its initial home position for precision of table while (digitalRead(tableCheck) == HIGH) Table.step(25); delay(500); Table.step(25); delay(500); //This loop is to nake the trigger start the table sequence if(Trigger.channell(0ool) == 1 && tableIndex <= tablePositions){ //This loop is to nake the winch and table work but only when the table is at home position //This loop is to nake the winch and table work but only when the table is at home position //This loop is to nake the winch and table work but only when the table is at home position //This loop is to nake the winch and table work but only when the table is at home position //This loop is to nake the winch and table work but only when the table is at home position //This loop is to nake the winch and table work but only when the table is at home position //This loop is to nake the winch and table work but only when the table is at home position //This loop is to nake the winch and table work but only when the table is at home position //This loop is to nake the winch and table work but only when the table is at home position //This code is to step the table to the appropriate distances in the tableCal matrix Table.step(-1*tableCal[tableIndex]); //This code is to step the table to the appropriate distances in the tableCal matrix Table.step(-1*tableCal[tableIndex]); //This code is imperative to the table preforms when moving the same steps back This code is imperative to the table to not miss the home position // Table.step(290*(tableIndex)-70); //This loop is to return and check if the table is at home position //This loop is to return and check if the table is at home position //This loop is to return and check if the table is at home position //This loop is to return and check if the table is at home position //This loop is to return and check if the table is at home position //This loop is to return and check if the table is at home position //This loop is to</pre>	179	
<pre>//Wove the table to its initial home position for precision of table while (digitalRead(tableCheck) == HIGH) Table.step(25); delay(500); // This if statement is to have the trigger start the table sequence if(Trigger.channel1(Bool) == 1 && tableIndex <= tablePositions){ // This lop is to make the winch and table work but only when the table is at home position while((winchIndex <= winchPositions) && (digitalRead(tableCheck) == 10W)){ // This lop is to make the winch and table work but only when the table is at home position while((winchIndex <= winchPositions) && (digitalRead(tableCheck) == 10W)){ // This lop is to make the winch and table work but only when the table is at home position while((winchIndex <= winchPositions) && (digitalRead(tableCheck) == 10W)){ // This code is to step the table to the appropriate distances in the tableCal matrix Table.step(-1*tableCal[tableIndex]); // This code is to step the table to the appropriate distances in the tableCal matrix Table.step(-1*tableCal[tableIndex]); // This code is in the table to the table preforms when moving the same steps back This code is inpertive to the table to not miss the home position // This loop is to return and check if the table is at home position */ Table.step(200*(tableIndex)-70); // This loop is to return and check if the table is at home position // This loop is to return and check if the table is at home position // This loop is to return and check if the table is at home position // This loop is to return and check if the table is at home position // This loop is to return and check if the table is at home position // This loop is to return and check if the table is at home position // This loop is to return and check if the table is at home position // This loop is to return and check if the table is at home position // This loop is to return and check if the table is at home position // This loop is to return and check if the table is at home position // This loop is to return and check if the table is at home position // This</pre>	180	
<pre>while (digitalRead(tableCheck) == HIGH)</pre>	181	//Move the table to its initial home position for precision of table
<pre>133 { 134 135 Table.step(25); 136 delay(500); 137 } 138 // This if statement is to have the trigger start the table sequence 139 if(Trigger.channel1(Bool) == 1 && tableIndex <= tablePositions){ 140 //This loop is to make the winch and table work but only when the table is at home position 151 while(winchIndex <= winchPositions) && (digitalRead(tableCheck) == LOM)){ 152 153 IPotor.writeMicroseconds(Trigger.channel3(MicroSeconds)); 154 rMotor.writeMicroseconds(Trigger.channel3(MicroSeconds)); 155 tableIndex++; 156 winchIndex*1560); 158 pumpAssembly(purgeTime); 159 delay(500); 150 //This code is to step the table to the appropriate distances in the tableCal matrix 150 Table.step(1*tableCal[tableIndex]); 164 delay(500); 175 pumpAssembly(sampleTime); 176 //This algorithm fixes the issue with poor signal quality from the moveTo function 177 It also fixes the issue of the drift the table preforms when avoing the same steps back 178 Table.step(290*(tableIndex)-70); 179 //This loop is to return and check if the table is at home position 170 while (digitalRead(tableCheck) == HIGH) 171 //This loop is to return and check if the table is at home position 172 while (digitalRead(tableCheck) == HIGH) 173 //This loop is to return and check if the table is at home position 171 //This loop is to return and check if the table is at home position 172 while (digitalRead(tableCheck) == HIGH) 173 //This loop is to return and check if the table is at home position 173 //This loop is to return and check if the table is at home position 174 //This loop is to return and check if the table is at home position 175 //This loop is to return and check if the table is at home position 175 //This loop is to return and check if the table is at home position 175 //This loop is to return and check if the table is at home position 175 //This loop is to return and check if the table is at home position 176 //This loop is to return and check if the table is at home position 177 //This loop is to return and check if the table is at home position</pre>	182	while (digitalRead(tableCheck) == HIGH)
<pre>184 185</pre>	183	{
185 Table.step(25); 186 delay(500); 187 } 188 // This if statement is to have the trigger start the table sequence 189 if(Trigger.channel1(Bool) == 1 && tableIndex <= tablePositions){	184	
<pre>delay(500);</pre>	185	Table.step(25);
<pre>187 } 187 // This if statement is to have the trigger start the table sequence 189 if(Trigger.channel1(Bool) == 1 && tableIndex <= tablePositions){ 190 //This loop is to nake the winch and table work but only when the table is at home position 191 while((winchIndex <= winchPositions) && (digitalRead(tableCheck) == LOM)){ 193 lMotor.writeMicroseconds(Trigger.channel3(MicroSeconds)); 194 rMotor.writeMicroseconds(Trigger.channel3(MicroSeconds)); 195 tableIndex++; 196 winchIndex++; 197 Winch.step(winchIndex*1500); 198 pumpAssembly(purgeTime); 199 200 201 delay(500); 202 //This code is to step the table to the appropriate distances in the tableCal matrix 203 Table.step(-1*tableCal[tableIndex]); 204 delay(500); 205 pumpAssembly(sampleTime); 206 /*This algorithm fixes the issue with poor signal quality from the moveTo function 207 It also fixes the issue of the drift the table preforms when moving the same steps back 208 This code is to return and check if the table is at home position 209 Table.step(290*(tableIndex)-70); 200 //This loop is to return and check if the table is at home position 201 while (digitalRead(tableCheck) == HIGH) 202 //This loop is to return and check if the table is at home position 203 while (digitalRead(tableCheck) == HIGH) 204 //This loop is to return and check if the table is at home position 205 while (digitalRead(tableCheck) == HIGH) 206 //This loop is to return and check if the table is at home position 207 while (digitalRead(tableCheck) == HIGH) 208 //This loop is to return and check if the table is at home position 209 //This loop is to return and check if the table is at home position 201 //This loop is to return and check if the table is at home position 202 //This loop is to return and check if the table is at home position 203 //This loop is to return and check if the table is at home position 204 //This loop is to return and check if the table is at home position 205 //This loop is to return and check if the table is at home position 206 //This loop is to return and check if</pre>	186	delay(500);
<pre>// This if statement is to have the trigger start the table sequence if(Trigger.Channel1(@ool) == 1 && tableIndex <= tablePositions){ //This loop is to make the winch and table work but only when the table is at home position while((winchIndex <= winchPositions) && (digitalRead(tableCheck) == 10w)){ whice(winchIndex <= winchPositions) && (digitalRead(tableCheck) == 10w)){ rMotor.writeMicroseconds(Trigger.Channel3(MicroSeconds)); tableIndex++; winchIndex++; winchIndex++; winchIndex++; winchIndex++; winchIndex++; winchIndex++; delay(500); delay(500); //This code is to step the table to the appropriate distances in the tableCal matrix Table.step(-1*tableCal[tableIndex]); delay(500); pumpAssembly(sampleTime); //This algorithm fixes the issue with poor signal quality from the moveTo function It also fixes the issue of the drift the table preforms when moving the same steps back This code is imperative to the table to not miss the home position */ Table.step(290*(tableIndex)-70); Table.step(290*(tableIndex)-70); //This loop is to return and check if the table is at home position while (digitalRead(tableCheck) == HIGH) //This loop is to return and check if the table is at home position while (digitalRead(tableCheck) == HIGH) //This loop is to return and check if the table is at home position while (digitalRead(tableCheck) == HIGH)</pre>	187	}
<pre>if(Trigger.Channel1(Bool) == 1 && tableIndex <= tablePositions){ //This loop is to make the winch and table work but only when the table is at home position while((winchIndex <= winchPositions) && (digitalRead(tableCheck) == LOW)){</pre>	188	<pre>// This if statement is to have the trigger start the table sequence</pre>
<pre>//This loop is to make the winch and table work but only when the table is at home position while((winchIndex <= winchPositions) && (digitalRead(tableCheck) == LOM)){ loop is to nwriteWicroseconds(Trigger.Channel3(MicroSeconds)); loop is tableIndex++; tableIndex++; loop is tableIndex]; loop is tableIndex]; loop is tableIndex+]; loop is tableInde</pre>	189	if(Trigger.Channel1(Bool) == 1 && tableIndex <= tablePositions){
<pre>while((winchIndex <= winchPositions) && (digitalRead(tableCheck) == LOW)){</pre>	190	//This loop is to make the winch and table work but only when the table is at home position
192 193 lMotor.writeMicroseconds(Trigger.Channel3(MicroSeconds)); 194 rMotor.writeMicroseconds(Trigger.Channel4(MicroSeconds)); 195 tableIndex++; 196 winchIndex++; 197 Winch.step(winchIndex*1500); 198 pumpAssembly(purgeTime); 199	191	while((winchIndex <= winchPositions) && (digitalRead(tableCheck) == LOW)){
193 IMotor.writeMicroseconds(Trigger.Channel3(MicroSeconds)); 194 rMotor.writeMicroseconds(Trigger.Channel4(MicroSeconds)); 195 tableIndex++; 196 winchIndex++; 197 Winch.step(winchIndex*1500); 198 pumpAssembly(purgeTime); 199	192	
194 rMotor.writeMicroseconds(Trigger.Channel4(MicroSeconds)); 195 tableIndex++; 196 winchIndex++; 197 Winch.step(winchIndex*1500); 198 pumpAssembly(purgeTime); 199	193	lMotor.writeMicroseconds(Trigger.Channel3(MicroSeconds));
195 tableIndex++; 196 winchIndex++; 197 Winch.step(winchIndex*1500); 198 pumpAssembly(purgeTime); 199	194	rMotor.writeMicroseconds(Trigger.Channel4(MicroSeconds));
196 winchIndex++; 197 Winch.step(winchIndex*1500); 198 pumpAssembly(purgeTime); 199	195	tableIndex++;
<pre>197 Winch.step(winchIndex*1500); 198 pumpAssembly(purgeTime); 199 200 201 delay(500); 202 //This code is to step the table to the appropriate distances in the tableCal matrix 203 Table.step(-1*tableCal[tableIndex]); 204 delay(500); 205 pumpAssembly(sampleTime); 206 /*This algorithm fixes the issue with poor signal quality from the moveTo function 207 It also fixes the issue of the drift the table preforms when moving the same steps back 208 This code is inperative to the table to not miss the home position */ 209 Table.step(290*(tableIndex)-70); 210 211 //This loop is to return and check if the table is at home position 212 while (digitalRead(tableCheck) == HIGH) 213 {</pre>	196	winchIndex++;
198 pumpAssembly(purgeTime); 199	197	Winch.step(winchIndex*1500);
<pre>199 200 201 delay(500); 202 //This code is to step the table to the appropriate distances in the tableCal matrix 203 Table.step(-1*tableCal[tableIndex]); 204 delay(500); 205 pumpAssembly(sampleTime); 206 /*This algorithm fixes the issue with poor signal quality from the moveTo function 207 It also fixes the issue of the drift the table preforms when moving the same steps back 208 This code is imperative to the table to not miss the home position */ 209 Table.step(290*(tableIndex)-70); 210 211 //This loop is to return and check if the table is at home position 212 while (digitalRead(tableCheck) == HIGH) 213 {</pre>	198	pumpAssembly(purgeTime);
200 201 delay(500); 202 //This code is to step the table to the appropriate distances in the tableCal matrix 203 Table.step(-1*tableCal[tableIndex]); 204 delay(500); 205 pumpAssembly(sampleTime); 206 /*This algorithm fixes the issue with poor signal quality from the moveTo function 207 It also fixes the issue of the drift the table preforms when moving the same steps back 208 This code is imperative to the table to not miss the home position */ 209 Table.step(290*(tableIndex)-70); 210 //This loop is to return and check if the table is at home position 211 //This loop is to return and check if the table is at home position 213 {	199	
201 delay(500); 202 //This code is to step the table to the appropriate distances in the tableCal matrix 203 Table.step(-1*tableCal[tableIndex]); 204 delay(500); 205 pumpAssembly(sampleTime); 206 /*This algorithm fixes the issue with poor signal quality from the moveTo function 207 It also fixes the issue of the drift the table preforms when moving the same steps back 208 This code is imperative to the table to not miss the home position */ 209 Table.step(290*(tableIndex)-70); 210 //This loop is to return and check if the table is at home position 212 while (digitalRead(tableCheck) == HIGH) 213 {	200	
<pre>202 //This code is to step the table to the appropriate distances in the tableCal matrix 203 Table.step(-1*tableCal[tableIndex]); 204 delay(500); 205 pumpAssembly(sampleTime); 206 /*This algorithm fixes the issue with poor signal quality from the moveTo function 207 It also fixes the issue of the drift the table preforms when moving the same steps back 208 This code is imperative to the table to not miss the home position */ 209 Table.step(290*(tableIndex)-70); 210 211 //This loop is to return and check if the table is at home position 212 while (digitalRead(tableCheck) == HIGH) 213 {</pre>	201	delay(500);
203 Table.step(-1*tableCal[tableIndex]); 204 delay(500); 205 pumpAssembly(sampleTime); 206 /*This algorithm fixes the issue with poor signal quality from the moveTo function 207 It also fixes the issue of the drift the table preforms when moving the same steps back 208 This code is imperative to the table to not miss the home position */ 209 Table.step(290*(tableIndex)-70); 210 //This loop is to return and check if the table is at home position 212 while (digitalRead(tableCheck) == HIGH) 213 {	202	<pre>//This code is to step the table to the appropriate distances in the tableCal matrix</pre>
204 delay(500); 205 pumpAssembly(sampleTime); 206 /*This algorithm fixes the issue with poor signal quality from the moveTo function 207 It also fixes the issue of the drift the table preforms when moving the same steps back 208 This code is imperative to the table to not miss the home position */ 209 Table.step(290*(tableIndex)-70); 211 //This loop is to return and check if the table is at home position 212 while (digitalRead(tableCheck) == HIGH) 213 {	203	Table.step(-1*tableCal[tableIndex]);
205 pumpAssembly(sampleTime); 206 /*This algorithm fixes the issue with poor signal quality from the moveTo function 207 It also fixes the issue of the drift the table preforms when moving the same steps back 208 This code is imperative to the table to not miss the home position */ 209 Table.step(290*(tableIndex)-70); 210 //This loop is to return and check if the table is at home position 212 while (digitalRead(tableCheck) == HIGH) 213 {	204	delay(500);
206 /*This algorithm fixes the issue with poor signal quality from the moveTo function 207 It also fixes the issue of the drift the table preforms when moving the same steps back 208 This code is imperative to the table to not miss the home position */ 209 Table.step(290*(tableIndex)-70); 210 //This loop is to return and check if the table is at home position 212 while (digitalRead(tableCheck) == HIGH) 213 {	205	<pre>pumpAssembly(sampleTime);</pre>
207 It also fixes the issue of the drift the table preforms when moving the same steps back 208 This code is imperative to the table to not miss the home position */ 209 Table.step(290*(tableIndex)-70); 210 //This loop is to return and check if the table is at home position 211 //This loop is to return and check if the table is at home position 212 while (digitalRead(tableCheck) == HIGH) 213 {	206	/*This algorithm fixes the issue with poor signal quality from the moveTo function
208 This code is imperative to the table to not miss the home position */ 209 Table.step(290*(tableIndex)-70); 210 //This loop is to return and check if the table is at home position 211 //This loop is to return and check if the table is at home position 212 while (digitalRead(tableCheck) == HIGH) 213 {	207	It also fixes the issue of the drift the table preforms when moving the same steps back
209 Table.step(290*(tableIndex)-70); 210	208	This code is imperative to the table to not miss the home position $*/$
210 211 //This loop is to return and check if the table is at home position 212 while (digitalRead(tableCheck) == HIGH) 213 {	209	Table.step(290*(tableIndex)-70);
211 //This loop is to return and check if the table is at home position 212 while (digitalRead(tableCheck) == HIGH) 213 {	210	
212 while (digitalRead(tableCheck) == HIGH) 213 {	211	<pre>//This loop is to return and check if the table is at home position</pre>
213 {	212	while (digitalRead(tableCheck) == HIGH)
	213	



Work in Progress and Future Work:

STRIDER can be further optimized by adding additional components as well as upgrades to the current infrastructure. GPS logging in conjunction with a water quality data package that includes pH, dissolved oxygen, and oxidation-reduction potential sensors will allow for the interpolation of data in a three-dimensional array for the water body. Re-manufacturing the paddlewheels to steel instead of aluminum will provide for a robust power delivery system. Implementing a flight

controller to open processing power will allow for more computing while allowing for STRIDER to become fully autonomous if necessary.

In the second version of STRIDER [See Figure 8], the body consists of a single green commercially available kayak modified with an aluminum superstructure to mount the water sampler, thruster, and electronics. Aluminum outriggers are attached for stability with floats that are mounted to the superstructure. The ISCO water sampler goes into the drum-shaped mount which provides a secure location for it to rest near the center of gravity of the boat. The thruster is mounted at the stern of the boat by an aluminum mount with a tube pulleyed to a servo mounted at the stern of the boat also. This provides pivotal steering for the Blue Robotics T100 Thruster at the bottom of the aluminum tube. A 12v lead-acid battery is mounted behind the ISCO. The electronics box is mounted behind the battery which houses the main processing unit and flight controller



Figure 8: Work in progress with new STRIDER

Trials and tribulations recognized from the first STRIDER implementation have been addressed and attempts have been made to remediate the issues in the second version through the implementation of commercialized systems integrated with the platform. Student teams could not get together during the winter break on campus but improvements were achieved in the early part of the ongoing spring semester by the new student team. In early March the team successfully ran a navigation test with the new STRIDER. Figure 9 are photographs taken during the successful navigation trial. The team intended to do a data collection run with both the old and new STRIDER platforms in March-April time frame, however, all project efforts have been put on hold due to COVID 19 related restrictions and social distancing mandates on campus for the time being.



Figure 9: Navigation field Test of New Strider with ISCO Sampler

Acknowledgment:

The primary author and the student members acknowledge the support from USDA Beltsville EMFS Laboratory, Maryland Space Grant, and the faculty/staff advisors at UMES.

Bibliography:

- Department of Environmental Quality, I. (2018). Common Water Quality Measures Idaho Department of Environmental Quality. Retrieved April 11, 2019, from <u>http://www.deq.idaho.gov/water-quality/surface-water/water-quality-criteria/common-measures/</u>
- 2. Gada, M. (n.d.). The Water Quality Test. Retrieved April 11, 2019, from <u>https://www.grc.nasa.gov/www/k-12/fenlewis/test.htm</u>
- Joshi, R., Derrickson, J., and Bane, N.J., Williams, M., and Nagchaudhuri, A., "Preliminary Trial Results for the Redesigned Strider Platform with Sampling Capability from Different Depths" Paper No. DETC2017-67385, pp. V009T07A049; 6 pages, Volume 9: 13th ASME/IEEE International Conference on Mechatronic and Embedded Systems and Applications, Cleveland, Ohio, USA, August 6–9, 2017
- Nagchaudhuri, A., Diab, A., Hartman, C., Mitra, M., Zhang, Lei, Pachepsky, Y., and Joshi. R., "Semi-Autonomous Tracking Robot with Instrumentation for Data-Acquisition and Environmental Research" Proceedings of 2016 Annual Conference and Exposition of American Society for Engineering Education, June 26-29, New Orleans, LA., 2016