Singaboat Technical Design Paper for the 2022 RobotX Competition

Du Jiajun, Greg A. G. Nonato, James Teo, Lim Jia Xing Gerald,
Liu Qiyuan, Ng Yu Fan Gerald, Wong See Kiat, Yun Yike

***Abstract*—Singaboat is a team from Nanyang Technological University, Singapore. Eight enthusiastic team members coming from different education background and nationality has been brought together by similar love in controls, automation and robotics to form a team attending RobotX Maritime Challenge. This paper shows the technical design and development of the Singaboat ASV.**

I. INTRODUCTION

NTU's team has participated to Maritime RobotX Challenge since its inception. A Wave Adaptive Modular Vessel (WAM-V 16) of the early version is used as the main mechanical platform, on which we have deployed: a) multiple-camera vision system, b) acoustic signal sensing system, c) on-vehicle controller, d) GPS system, e) propulsion system, f) battery system, etc. Singaboat is a team including 7 undergraduates and 1 PhD student, with guidance from technical advisor and assistance of technical staffs from Innovation lab, Nanyang Technological University.

II. DESIGN STRATEGY

Compared to the last RobotX challenge 2018, team decides to use MATLAB as main software platform instead of programming on ROS to achieve the various tasks. For the hardware, propulsion system, vision system and processing units have to be upgraded to achieve a better performance. Team is separated into 3 major groups, perception, planning, and control. Each different group works on both ASV and UAV.

A. Mechanical Design

1. ASV

The team decided to re-use most of the existing components which come from the previous batches to save our time. However, as the team plan to operate the ASV via MATLAB, some components cannot be re-used, such as central processing unit, a laptop is chosen to replace the embedded PC and mounted onto the ASV. Another advantage is to save the power consumption from the power supply, as the laptop can run without power supply and last 2-3 hours per charge. Due to time constraints, no formal waterproofing is designed, and this issue will be fixed after the competition.

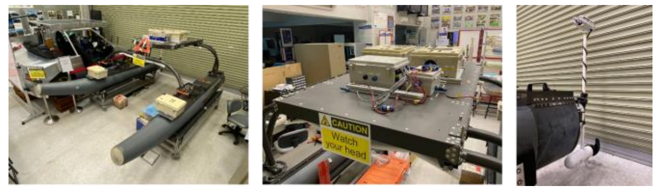

Figure 1: General Mechanical Design for the ASV

a. Camera Specification: **Axis m2014-e network camera.**
IP camera is a type of digital video camera that receives control data and sends image data via an IP network. Once connected to the network, it can fetch video signal remotely from ASV and load vision input to data processing server. It plays a key role in navigation and recognition.

b. GPS Specification: **Hemisphere R330 multi-GNSS receiver.**
GPS is able to receive geographical data about the location of the craft, in order to know where the craft is at every given time. The GPS consists of 2 main components, namely the receiver and the antenna.
The receiver of choice is the R632GNSS Receiver by Hemisphere. The receiver is compact and powerful. It boasts the ability to have a heading accuracy of up to 0.01 degrees. It offers interference rejection and multipath mitigation. The receiver also offers multiple methods of connectivity and an array of wireless communications. It offers worldwide stand-alone positioning to 4cm.

c. Antenna Specification: **Hemisphere A43 Antenna.**
Then antenna is a device that receives radio signals from the GPS satellites at varying frequencies. The antenna then converts these signals into electrical signals which can be interpreted using a GPS receiver to calculate the receiver's location.
The A43 Antenna is a multi-GNSS precision antenna. The antenna is able to be used in challenging environments such as near buildings and foliage. It has superior multipath mitigation, stable phase center and strong SNR's even at low elevations.

d. Hydrophone Specification: **Teledyne Reson TC4013.**

Teledyne Reson TC4013 has the following properties: High sensitivity, Omnidirectional to high frequencies, Broad banded, O-ring sealed mounting, Individually calibrated.

e. Motor Specification: **Minn Kota RT80.**
Two Minn Kota RT80 propellers were attached at the right and left end of the vessel to propel the vessel forward or backward. Minn Kota RT80 propeller requires 24V and able to draw up to a maximum of 56A.
Two Arduino Uno are needed to power the propeller individually by taking in input of velocity from planning group and convert it to produce voltage output.
Two SparkFun I2C DAC Breakout – MCP4725 is a low-powered, high accuracy, single channel, 12-bit voltage output Digital-to-Analog Convertor (DAC). The DAC comes with non-volatile memory which enables the DAC device to hold the DAC input code during power-off time and the DAC will update the code immediately after power up. SparkFun I2C DAC Breakout – MCP4725 will be individually attached to an Arduino to produce an output of 0 – 5V to the signal of the propeller.

2. UAV

The UAV will be mounted with a camera and robot arm to support the ASV by accomplishing the following tasks: 1. Area mapping 2. Object detection 3. Object pick and place. DJI F550 Flamewheel is used as main chassis for the UAV. 2212 Brushless DC motor 920kv is chosen to drive the propeller and 30A ESC speed controller is used to regulate the speed of motors.

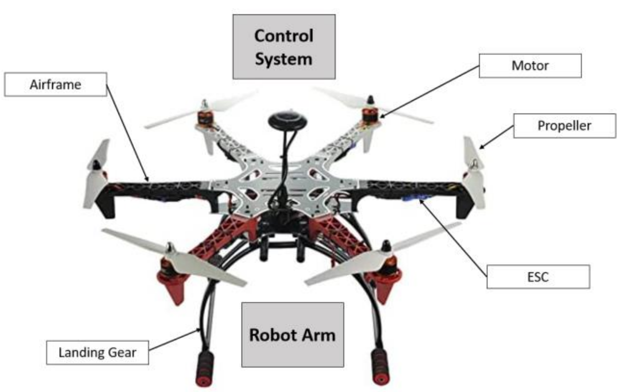

Figure 2: General Mechanical Design for the UAV

Component	Model	Description
Airframe	DJI F550 Flamewheel	Main Chassis for the UAV
Landing Gear	F550 Landing Gear	
Propeller	1045 Propellers	Pushes air to produce thrust
Motor	2212 Brushless DC motor 920kv	Spins the propeller to produce thrust
ESC	30A ESC	Regulates speed of motors, with inputs from the flight controller

Figure 3: Hardware List for Drone

Component	Model	Description
Flight Controller	Pixhawk 2.4.8	Controls the speed of the motors based on the information obtained from its sensors
Companion Computer	Raspberry Pi 3	Performs complex autonomous functions
GPS module	NEO-M8N GPS	Provides GPS coordinates to the flight controller
Radio Telemetry Module	3DR 915Mhz Radio Wireless Telemetry Set	Provides radio communication between the flight controller and the ground control station
Battery	Onbo 2850mAh 60C 3S Lipo Battery	Powers the drone
Power Distribution Board	Sky-Drones SmartAP Power Distribution Board	Provides varying voltage sources for the different electrical modules on the UAV
Camera	Arducam Stereo Camera	Provides visual inputs to the companion computer

Figure 4: Hardware List for Control System

Component	Model	Description
Servo Driver	PCA9685 16 Channel 12 Bit PWM Servo Driver	Allows microcontroller to drive several servo motors with I2C
Servos	Hitec HS311 Standard Servo 3.5Kg 180°	3.5kg.cm torque Servo motor
	DSS-M15 DF Metal Geared Standard Servo 15Kg 180°	13.5kg.cm torque Servo motor
	DF05BB Standard Servo 5Kg	4.8kg.cm torque Servo motor

Figure 4: Hardware List for Robot Arm

B. Electrical Design

1. ASV

Physical connection between the various components showing the communication and wiring. Overall diagram is as follows:

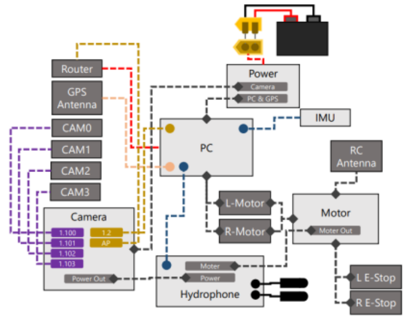

Figure 5: Overall Wiring Diagram

The various boxes will connect and communicate. Each of the green dash boxes is a physical waterproof box that houses the connection. The red box with white text is the physical components that come out of the box to interact with the real world.

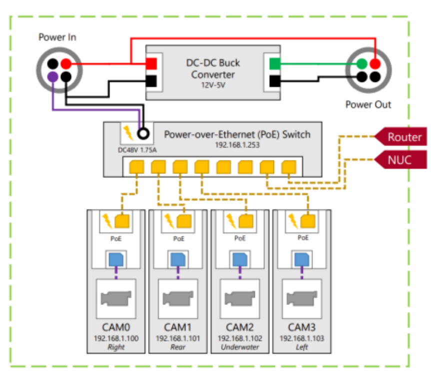

Figure 6: Wiring Diagram for Camera

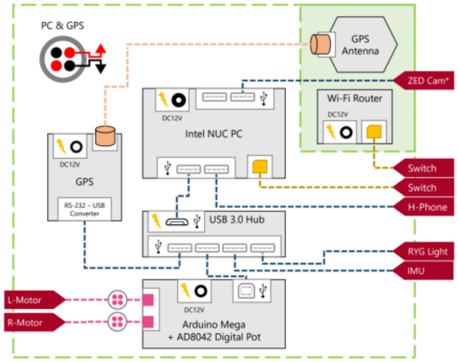

Figure 7: Wiring Diagram for GPS

For the motor, the computer will determine the necessary thrust required of the two motors individually. The required inputs will be received by the Arduino which are connected to each motor. The Arduino will do the necessary calculations and produce the required signals to control the speed and thrust of the motors.

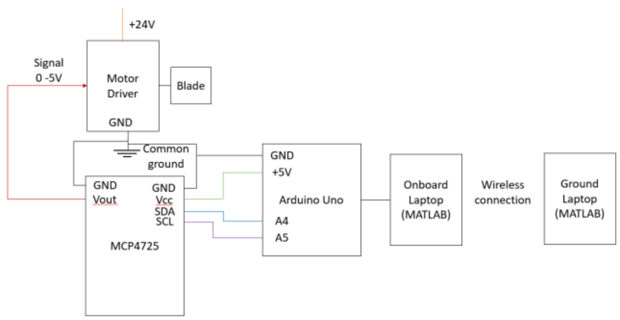

Figure 8: Wiring Diagram for Motor

2. *UAV*

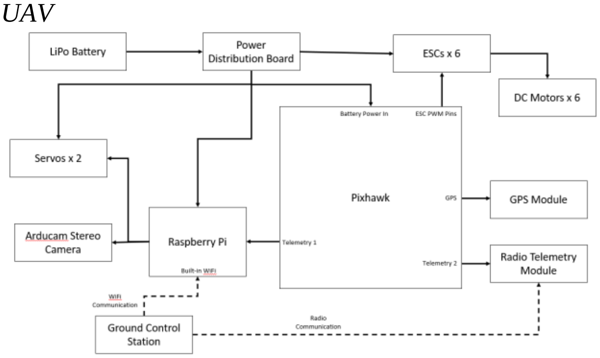

Figure 9: Control System Connection Diagram (1)

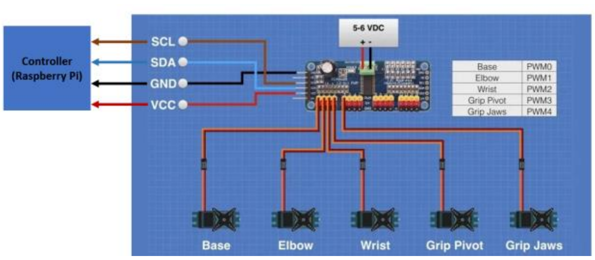

Figure 10: Control System Connection Diagram (2)

C. Software Design

1. *ASV*

a. Vision Based perception system:

The input is the video signal from IP camera. And output is visual coordinate map & identity of objects.

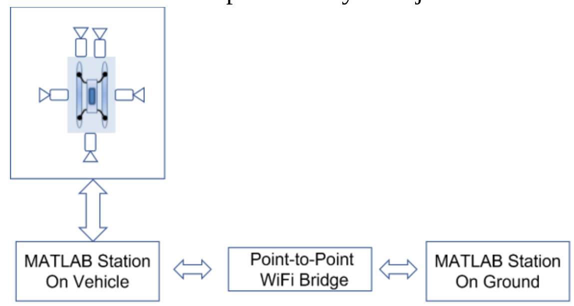

Figure 11: Perception System

To connect the IP camera to our local computational server, an Visual Perception API (Advance Software Interface) must be implemented to remotely connect hardware and software system.

Once onboard IP camera successfully connected to a public network, it will be assigned an unique URL (Uniform Resouces Locator). This API will try directly access the onboard IP camera by connecting to that URL. After connection is stabilized, the real time video signal can be fetched back for later usages.

Video Signal Pre-Processing:

The real time IP camera video signal can be very noisy when received by server, hence this pre-processing of video signal input is necessary for a smooth processing later. The noises can arise for a lot of reasons: different lighting conditions, semi-periodical vibration caused by sea waves, reflections of light on sea surface, data losses in transfer and etc.

This pre-processing program will freeze video signal every 0.1 seconds to get an image. Each image will be first scanned to generate a corresponding histogram, which will be used to analyze the pixel value distribution to see if there are any needs to redistribute the histogram for easier processing. Since our target objects are all of regular shape and color, a noise filter will be cast on adjusted images to remove all noise points. So that the remaining image will no longer have noise image that will slow done later recognition process.

2D Camera Frame to World Frame Geometry Transformation:

To navigate our ASV on sea based on visual perception system, we must obtain a transformation matrix that being able to find the correspondence between camera and world frame points.

This process makes use of the geometry relationship between camera frame coordinates and world frame coordinates, to compute a transformation matrix via calibration. When entire vision system is brought online, the program will first freeze one image for calibration. To achieve high accuracy, user will select 4meaningful pixel points in this image by manually clicking onto it. Later the program will ask user to input the corresponding world frame points of the previous selected 4 pixel points. The transformation matrix is then determined. Note that: as in the scene & task the height is not important, so the program discard height computation to increase its reaction speed. Once the transformation matrix is obtained, it will automatically apply on each point of the image to obtain world geometry information of each pixel. The real-world coordinates will share the same frame which is exact at the center of the boat.

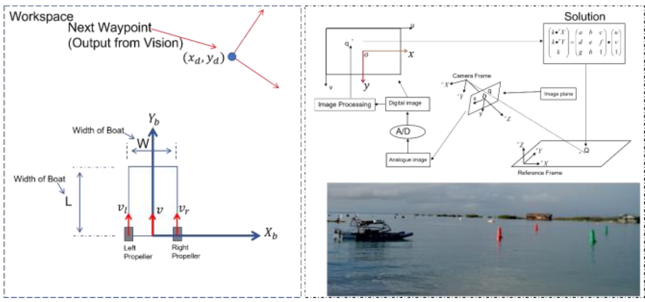

Figure 12: ASV Navigation

Spherical Color Identity Analysis:
Since the different tasks are defined by the buoy color, we need a color recognition system to be able to fast identify the color information under various lighting conditions. And it needs to classify the recognized color to its corresponding tasks.
This program can be divided into 2 parts: Color Sphere Building and Color Recognition. The first part, Color Sphere Building, is a leaning part for our program to learn what to recognize under human supervision. And the second part, Color Recognition, is to apply what has been learned to recognize the target object.

Color Sphere Learning:
The program will first freeze several images and ask user to click as many points as possible that can represent the object. Then based on the selected RGB values, the mean average and standard deviation is calculated. Since the RGB domain can be treated as a 3-dimensional space, if take mean average as a center point and standard deviation as radius, a color sphere can be constructed. Theoretically, with more and more training, the program will progressively learn from user to refine the sphere. This algorithm makes use of REC Neuron Network, which is exactly how humans learn new things: we progressively learn about a new event and keep adjusting our cognition to that event.

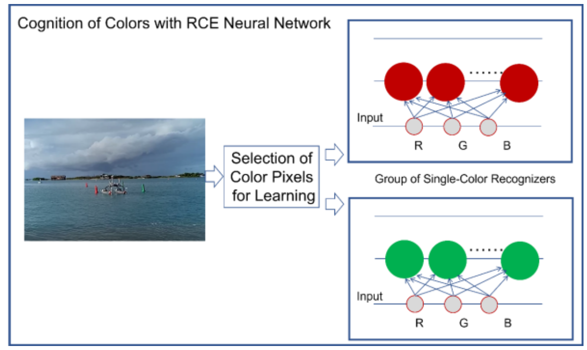

Figure 13: Colour Cognition

Color Recognition:
After each recognition sphere is constructed, the standard spherical color data base is fully constructed. The new input image will be scanned on a pixel level to check if the RGB value of that pixel falls into any sphere. And in the meantime, a template matching method is used to first extract the object and give it a label. If the same object of same label falls into same sphere for at least 10 images, then it can be classified into that color class.

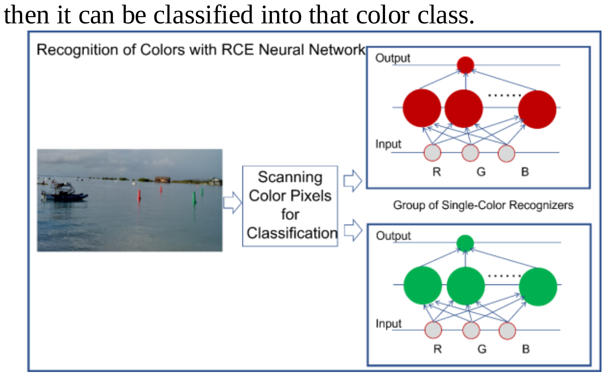

Figure 14: Colour Recognition

Template Matching:
For our ASV to be able to quickly recognize the object and response to it while navigating, the template matching is needed to first identify the pixel area of object to ease the workload. Matlab template match app will generate this program automatically.

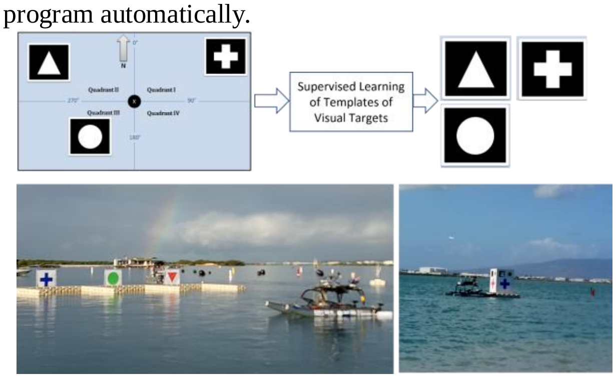

Figure 15: Template Matching (1)

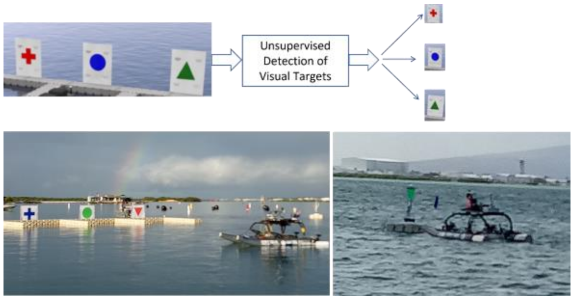

Figure 16: Template Matching (2)

b. *GPS:*
The input is satellite data. And output is latitude, longitude, heading, velocity, these data points will be passed on to the planning group.
The software is needed to convert the data received by the GPS receiver into values and data for the planning team.

c. *Hydrophone:*
The input is digital data of acoustic signals. And output is frequencies of acoustic signals, and presence of sound emitter of interest.

d. *Planning system:*
The goal is to be able to achieve output velocity of left and right motors from the input waypoint. Destination coordinates or by vision will be given by perception team and transmit the next waypoint to planning group. A ground computer with MATLAB will take in the waypoint and output left and right motors velocity.
The input is next waypoint which is the destination that the vessel will have to arrive at. And output is the velocity of left and right motors to allow the vessel to move towards the destination.
After receiving the waypoint from the perception group and locating it on the output display, the angular velocity (ω) will be able to be obtained by assuming the vessel follows a planned circular path with the radius (R) and moving at a circular velocity (v). $\omega = \frac{v}{R}$
In theory, many solutions determine the circular path radius (R). We have decided to proceed with the simplest version of obtaining the circular path radius (R). We will get the waypoint coordinates (xd, yd) from the perception group and using the formula 1/xd; we will be able to calculate the radius (R). For instance, if the xd value is 0, the R-value will be infinite; hence the vessel will move in a straight line. $R = \frac{1}{x_d}$
Assuming circular velocity(v) is at the middle of both left, and right propellers and both propellers are moving at a circular velocity (vl and vr). We can calculate left and right propeller velocity with the known value of radius(R) and angular velocity (ω). The circular velocity of the left propeller: $v_l = \frac{v}{R}\left(R + \frac{W}{2}\right) = v(1 + \frac{W}{2R})$ and the velocity of the right propeller: $v_r = \frac{v}{R}\left(R - \frac{W}{2}\right) = v(1 - \frac{W}{2R})$.

e. *Control System:*
The software would control the received input velocity of left and right and output required voltage for the maneuvering of the vessel.
Using Arduino with MCP4725 to program the input velocity from the planning group and convert it to 12bit binary numbers before converting it to output voltage. Converting the velocity to binary number will allow small increment of voltage, thus allowing small adjustment in the voltage output. When 0V – 2.4V are sent into the motor, it will reverse the propellers, and 2.6V - 5V will enable the propellers to move forward. When exact 2.5V is sent in, it will stop the propellers from spinning.

Output Voltage	Propeller's movement
0V – 2.4V	Backward throttle
2.5V	No spin on engines
2.6V - 5V	Forward throttle

Figure 17: Control System (1)

For a 12bit DAC, it will contain 4096 steps (including 0) and with a limit of 5V, 1.2mV increment of voltage for each step. Assuming the minimum and maximum velocity of the motor to be –20km/h to 20km/h, any value given below or above the limit will be converted to the minimum or maximum amount. When a negative velocity is given into the Arduino, an absolute value will be given to the velocity as there is no negative binary count. To ensure that there is no confusion, an if else statement will be used to change the value outcome by adding or subtracting 2.5V thus allowing the vessel to be able to do forward or backward motion.

Binary steps	Input velocity	Output voltage
4096	20km/h	5V
2048	0km/h	2.5V
0	-20km/h	0V

Figure 17: Control System (2)

f. Demonstration System:
This system shows users the outputs of each system and locate the vessel away from the ground computer. Able to know if any errors are occurring on each system and quickly rectify them.
Using MATLAB app designer to create a GUI display on the ground computer and allow the user to input data to be transmitted to onboard NUC via Wi-Fi connection. Under the test window, there will be perception, planning and control buttons to create a popup window when clicked. The run window will allow users to create a popup window for each task when clicked. The green light will be placed on the top right-hand corner of the GUI to indicate the Wi-Fi connection and will display a red light when the connection is lost. Vehicle and drone on/off buttons are placed at the bottom to allow the user to start and stop without terminating the connection. The exit button will enable the user to terminate the connection between the ground and host computers.

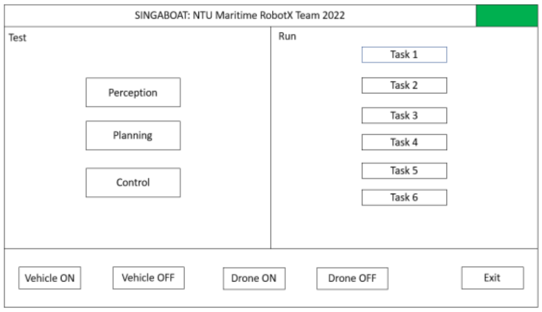

Figure 18: Demonstration System (1)

The green light will be placed on the top right-hand corner of the GUI to indicate the Wi-Fi connection and will display a red light when the link is lost. The perception group consists of GPS, Compass, Acoustic and Vision to be input by hardware on the left. The display on the right is implemented by getting the limit of the area (x1, y1), (x2, y2), (x3, y3), and (x4, y4) from user input and using 2D vision matrix calculations to achieve the object plane (u1, v1), (u2, v2), (u3, v3) and (u4, v4). Destination coordinates were also converted to (xd, yd), which will appear in the output. Start/Stop buttons are placed on the bottom to allow the user to start and stop it without terminating the connection. The exit button will enable the user to remove the group link.

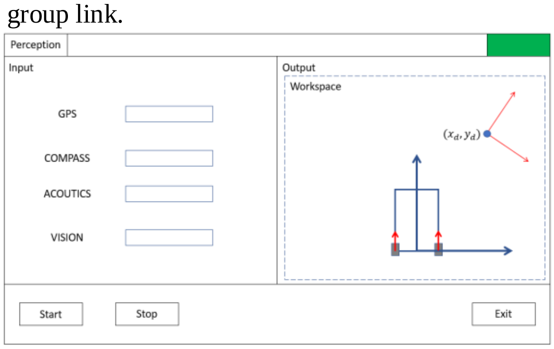

Figure 19: Demonstration System (2)

The green light will be placed on the top right-hand corner of the GUI to indicate the Wi-Fi connection and will display a red light when the link is lost. The planning group obtains its information from the perception group, which includes GPS and Destination. The display on the right is implemented by getting the limit of the area (x1, y1), (x2, y2), (x3, y3), and (x4, y4) from user input (planning group) and using 2D vision matrix calculations to achieve the object plane (u1, v1), (u2, v2), (u3, v3) and (u4, v4). Destination coordinates were also converted to (xd, yd), which will appear in the output. Start/Stop buttons are placed on the bottom to allow the user to start and stop it without terminating the connection. The exit button will enable the user to remove the group link.

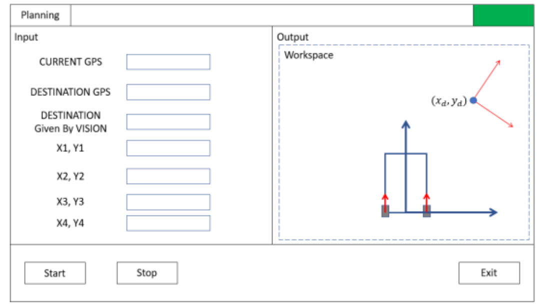

Figure 20: Demonstration System (3)

The green light will be placed on the top right-hand corner of the GUI to indicate the Wi-Fi connection and will display a red light when the link is lost. The control group will obtain left and right motor speed from the planning group and convert it into velocity. The display on the right is implemented by getting the limit of the area (x1, y1), (x2, y2), (x3, y3), and (x4, y4) from user input and using 2D vision matrix calculations to achieve the object plane (u1, v1), (u2, v2), (u3, v3) and (u4, v4). Destination coordinates were also converted to (xd, yd), which will appear in the output. Start/Stop buttons are placed on the bottom to allow the user to start and stop it without terminating the connection. The exit button will enable the user to remove the group link.

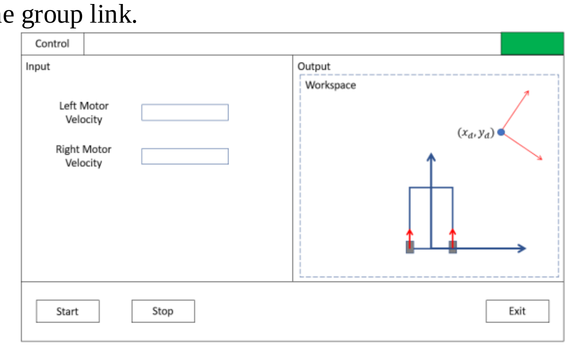

Figure 21: Demonstration System (4)

2D vision matrix was used to calculate the location of destination coordinates by input the x and y axis limit and U and V display output limits.

$$\begin{pmatrix} a & b & c \\ d & e & f \\ g & h & 1 \end{pmatrix}\begin{pmatrix} x1 \\ y1 \\ 1 \end{pmatrix} = \begin{pmatrix} sU1 \\ sV1 \\ s \end{pmatrix},$$

$$\begin{pmatrix} a & b & c \\ d & e & f \\ g & h & 1 \end{pmatrix}\begin{pmatrix} x2 \\ y2 \\ 1 \end{pmatrix} = \begin{pmatrix} sU2 \\ sV2 \\ s \end{pmatrix},$$

$$\begin{pmatrix} a & b & c \\ d & e & f \\ g & h & 1 \end{pmatrix}\begin{pmatrix} x3 \\ y3 \\ 1 \end{pmatrix} = \begin{pmatrix} sU3 \\ sV3 \\ s \end{pmatrix} \text{ and}$$

$$\begin{pmatrix} a & b & c \\ d & e & f \\ g & h & 1 \end{pmatrix}\begin{pmatrix} x4 \\ y4 \\ 1 \end{pmatrix} = \begin{pmatrix} sU4 \\ sV4 \\ s \end{pmatrix}.$$

With the x, y, U and V limits, the users will be able to calculate the eight unknows a, b, c, d, e, f, g and h by making U and V the subject.

$$\begin{cases} sU_1 = ax_1 + by_1 + c \\ sV_1 = dx_1 + ey_1 + f \\ s = gx_1 + hy_1 + 1 \end{cases} \rightarrow$$

$$U1 = \frac{ax_1+by_1+c}{gx_1+hy_1+1},\ V1 = \frac{dx_1+ey_1+f}{gx_1+hy_1+1} \rightarrow$$
$$U1 = ax1 + by1 + c - gU1x1 - hU1y\ ,$$
$$V1 = dx1 + ey1 + f\ - gV1x1 - hV1y.$$

$$\begin{cases} sU_2 = ax_2 + by_2 + c \\ sV_2 = dx_2 + ey_2 + f \\ s = gx_2 + hy_2 + 1 \end{cases} \rightarrow$$

$$U_2 = \frac{ax_2+by_2+c}{gx_2+hy_2+1},\ V_2 = \frac{dx_2+ey_2+f}{gx_2+hy_2+1} \rightarrow$$
$$U_2 = ax_2 + by_2 + c\ - gU_2x_2 - hU_2y_2,$$
$$V_2 = dx_2 + ey_2 + f\ - gV_2x_2 - hV_2y$$

$$\begin{cases} sU_3 = ax_3 + by_3 + c \\ sV_3 = dx_3 + ey_3 + f \\ s = gx_3 + hy_3 + 1 \end{cases} \rightarrow$$

$$U_3 = \frac{ax_3+by_3+c}{gx_3+hy_3+1},$$
$$V_3 = \frac{dx_3+ey_3+f}{gx_3+hy_3+1} \rightarrow$$
$$U_3 = ax_3 + by_3 + c\ - gU_3x_3 - hU_3y_3,$$
$$V_3 = dx_3 + ey_3 + f\ - gV_3x_3 - hV_3y$$

$$\begin{cases} sU_4 = ax_4 + by_4 + c \\ sV_4 = dx_4 + ey_4 + f \\ s = gx_4 + hy_4 + 1 \end{cases} \rightarrow$$
$$U_4 = \frac{ax_4+by_4+c}{gx_4+hy_4+1},\ V_4 = \frac{dx_4+ey_4+f}{gx_4+hy_4+1}$$
$$U_4 = ax_4 + by_4 + c\ - gU_4x_4 - hU_4y_4,$$
$$V_4 = dx_4 + ey_4 + f\ - gV_4x_4 - hV_4y_4$$

With individual U and V value as the subject, the user will be able to combine all the matrix together to obtain the eight equation and eight unknowns. With the following formula, $Z = (A^tA)^{-1}A^tB$, and the bottom equation the user will be able to achieve the eight unknowns value thus able to obtain the destination coordinates.

$$\begin{bmatrix} x1 & y1 & 1 & 0 & 0 & 0 & -U1x1 & -U1y1 \\ 0 & 0 & 0 & x1 & y1 & 1 & -V1x1 & -V1y1 \\ x2 & y2 & 1 & 0 & 0 & 0 & -U2x2 & -U2y2 \\ 0 & 0 & 0 & x2 & y2 & 1 & -V2x2 & -V2y2 \\ x3 & y3 & 1 & 0 & 0 & 0 & -U3x3 & -U3y3 \\ 0 & 0 & 0 & x3 & y3 & 1 & -V3x3 & -V3y3 \\ x4 & y4 & 1 & 0 & 0 & 0 & -U4x4 & -U4y4 \\ 0 & 0 & 0 & x4 & y4 & 1 & -V4x4 & -V4y4 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \\ e \\ f \\ g \\ h \end{bmatrix} = \begin{bmatrix} U1 \\ V1 \\ U2 \\ V2 \\ U3 \\ V3 \\ U4 \\ V4 \end{bmatrix}$$

$$A = \begin{bmatrix} x1 & y1 & 1 & 0 & 0 & 0 & -U1x1 & -U1y1 \\ 0 & 0 & 0 & x1 & y1 & 1 & -V1x1 & -V1y1 \\ x2 & y2 & 1 & 0 & 0 & 0 & -U2x2 & -U2y2 \\ 0 & 0 & 0 & x2 & y2 & 1 & -V2x2 & -V2y2 \\ x3 & y3 & 1 & 0 & 0 & 0 & -U3x3 & -U3y3 \\ 0 & 0 & 0 & x3 & y3 & 1 & -V3x3 & -V3y3 \\ x4 & y4 & 1 & 0 & 0 & 0 & -U4x4 & -U4y4 \\ 0 & 0 & 0 & x4 & y4 & 1 & -V4x4 & -V4y4 \end{bmatrix}$$

$$B = \begin{bmatrix} U1 \\ V1 \\ U2 \\ V2 \\ U3 \\ V3 \\ U4 \\ V4 \end{bmatrix} \quad Z = \begin{bmatrix} a \\ b \\ c \\ d \\ e \\ f \\ g \\ h \end{bmatrix}$$

g. *Task 1:*

The green light will be placed on the top right-hand corner of the GUI to indicate the Wi-Fi connection and will display a red light when the link is lost. On the left-hand side of the GUI, show the button to toggle ON/OFF individual systems, as some functions are not required. The output on the right shows the vessel location, heading, and destination given by vision or coordinates. Start/Stop buttons are placed on the bottom to allow the user to start and stop it without terminating the connection. The exit button will enable the user to remove the group link.

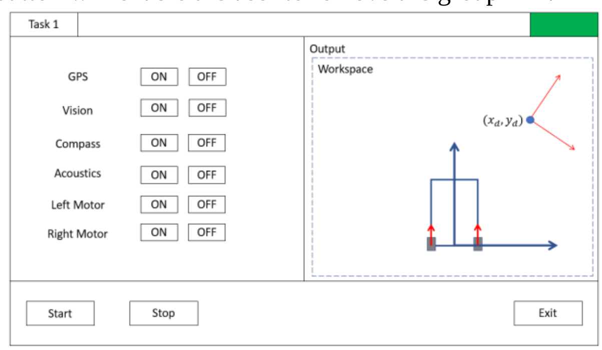

Figure 22: Design of GUI (1)

A set of three gates will be located in the course area with a beacon placed underwater within each gate. The AMS must detect the active underwater beacon, transit through the gate in which the active beacon is located, and then circle one of two buoys. The Beacon to be used is the Teledyne Benthos ALP-365 Pinger used in previous years. Our proposed solution is described in the form of the flowchart as shown in Figure 23:

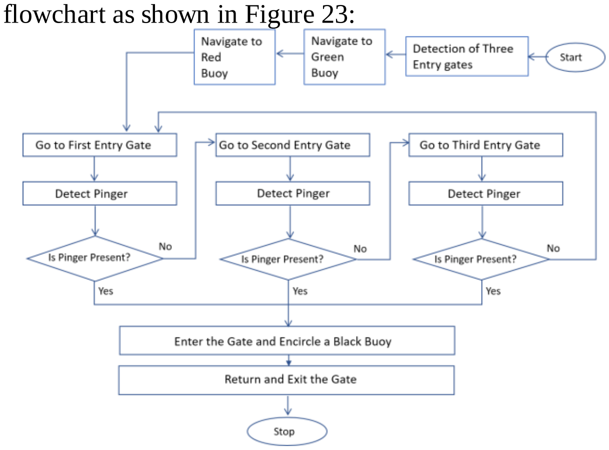

Figure 23: Solution of Task 1

h. *Task 2:*

The green light will be placed on the top right-hand corner of the GUI to indicate the Wi-Fi connection and will display a red light when the link is lost. On the left-hand side of the GUI, show the button to toggle ON/OFF individual systems, as some functions are not required. The output on the right shows the vessel location, heading, and destination given by vision or coordinates. Start/Stop buttons are placed on the bottom to allow the user to start and stop it without terminating the connection. The exit button will enable the user to remove the group link.

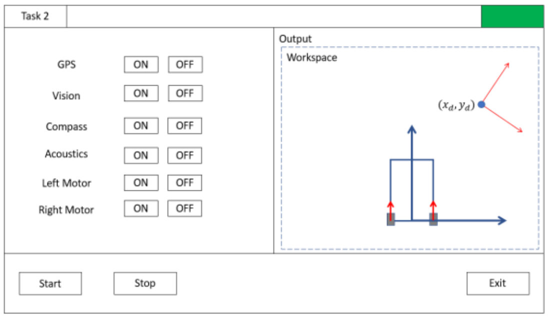

Figure 24: Design of GUI (2)

This task is modelled after the “Traverse Navigation Channel” task from the 2019 Virtual RobotX competition. The AMS must deploy a UAV to map the challenge task, and then use this to guide the WAM-V through a path defined by sets of buoys, where each set is a pair of red/green coloured buoys. The exact buoy types may vary from what was used in the Virtual RobotX competition. Obstacles may be included within and around the path.
Our solution is described in the form of the flowchart as shown in Figure 25:

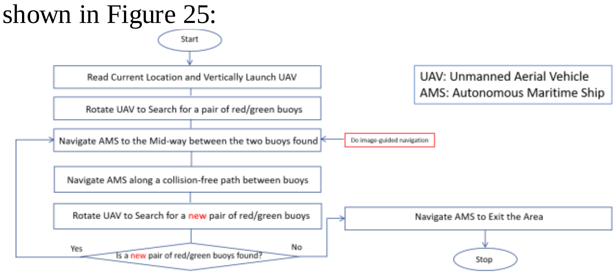

Figure 25: Solution of Task 2

i. *Task 3:*

The green light will be placed on the top right-hand corner of the GUI to indicate the Wi-Fi connection and will display a red light when the link is lost. On the left-hand side of the GUI, show the button to toggle ON/OFF individual systems, as some functions are not required. The output on the right shows the vessel location, heading, and destination given by vision or coordinates. Start/Stop buttons are placed on the bottom to allow the user to start and stop it without terminating the connection. The exit button will enable the user to remove the group link.

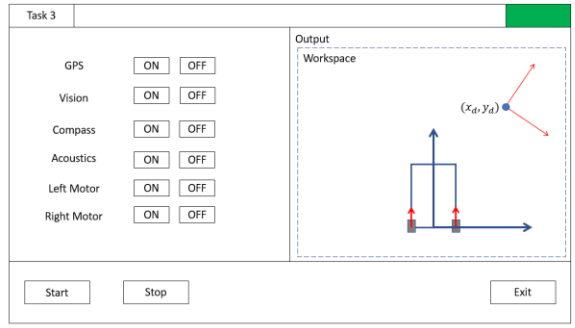

Figure 26: Design of GUI (3)

The Wildlife Encounter and Avoid task requires the UAV to identify objects of interest and inform the WAM-V to circumnavigate the object. These objects of interest will represent Australian Marine Life such as platypi, turtles and crocodiles and will each be painted in distinct coatings with unique spectral signatures to enable identification and classification with a hyperspectral camera. To successfully circle the marine life, the AMS must transit around them until it has crossed its original path, transiting at least 360 degrees. The clockwise/counter-clockwise direction will be based on the classification of the marine life by their spectral signatures (to be developed).
Our solution is described in the form of the flowchart as shown in Figure 27:

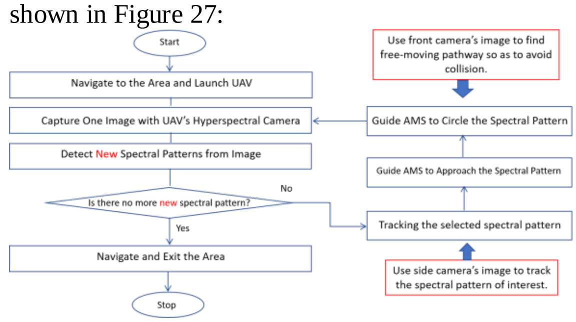

Figure 27: Solution of Task 3

j. *Task 4:*

The green light will be placed on the top right-hand corner of the GUI to indicate the Wi-Fi connection and will display a red light when the link is lost. On the left-hand side of the GUI, show the button to toggle ON/OFF individual systems, as some functions are not required. The output on the right shows the vessel location, heading, and destination given by vision or coordinates. Start/Stop buttons are placed on the bottom to allow the user to start and stop it without terminating the connection. The exit button will enable the user to remove the group link.

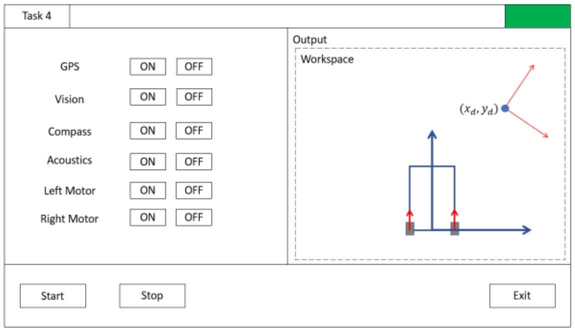

Figure 28: Design of GUI (4)

The Scan the Code task has been a staple of the RobotX Challenge, since the inaugural event in 2014. The AMS is required to observe a light sequence displayed by an RGB buoy and report the colour pattern. The light assembly on the buoy will successively display colours one at a time to generate a sequential pattern of three colours (e.g. red-green-red).

Our proposed solution is described in the form of the flowchart as shown in Figure 29:

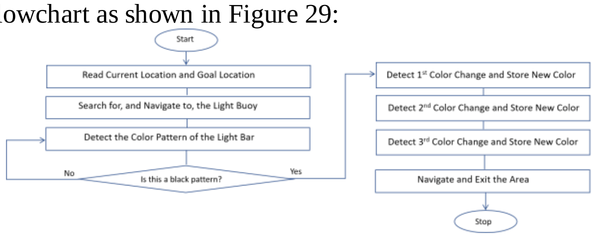

Figure 29: Solution of Task 4

k. *Task 5:*

The green light will be placed on the top right-hand corner of the GUI to indicate the Wi-Fi connection and will display a red light when the link is lost. On the left-hand side of the GUI, show the button to toggle ON/OFF individual systems, as some functions are not required. The output on the right shows the vessel location, heading, and destination given by vision or coordinates. Start/Stop buttons are placed on the bottom to allow the user to start and stop it without terminating the connection. The exit button will enable the user to remove the group link.

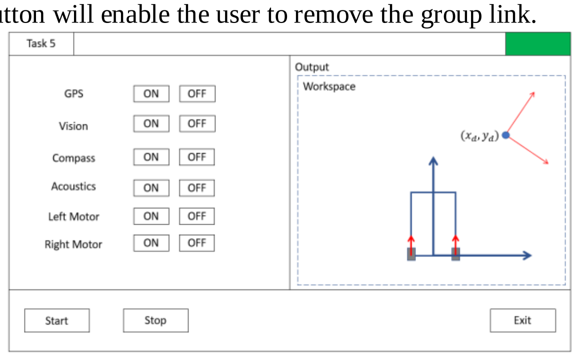

Figure 30: Design of GUI (5)

For the 2022 RobotX Challenge, the docking bays will have a parallel dock configuration (similar to the 2016 RobotX Challenge). This task combines the docking task and the Detect and Delivery task from 2018 RobotX Challenge. The Dock and Delivery task will be anchored in the course. The AMS will need to dock in the bay displaying the correct coloured light. Once docked, the AMS will deliver a payload (racquetball) into one of the holes (located above the coloured light). As in previous years, there will be a smaller and a larger hole for payload delivery.

Our solution is described in the form of the flowchart as shown in Figure 31:

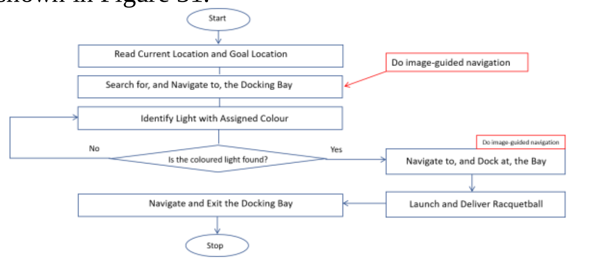

Figure 31: Solution of Task 5

l. *Task 6:*

The green light will be placed on the top right-hand corner of the GUI to indicate the Wi-Fi connection and will display a red light when the link is lost. On the left-hand side of the GUI, show the button to toggle ON/OFF individual systems, as some functions are not required. The output on the right shows the vessel location, heading, and destination given by vision or coordinates. Start/Stop buttons are placed on the bottom to allow the user to start and stop it without terminating the connection. The exit button will enable the user to remove the group link.

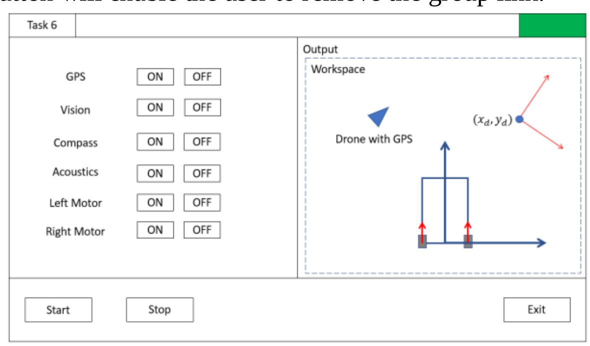

Figure 32: Design of GUI (6)

This is a new task for the 2022 RobotX Challenge which will use the UAV to pick up an item from the dock and deliver it to a helipad ashore.

Our proposed solution is described in the form of the flowchart as shown in Figure 33:

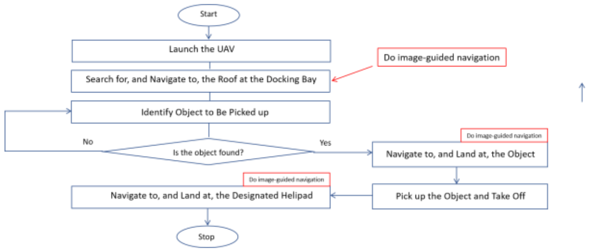

Figure 33: Solution of Task 6

D. Acknowledgements

The team is fully sponsored by Future Systems and Technology Directorate (FSTD), Ministry of Defense (MINDEF)， Singapore. We appreciate the funding support.

The team would also like to appreciate ST Engineering Ltd. Co. to provide a testing site on sea area in Singapore.