Utilization of Autonomous Navigation and Path Planning for Chemical Processes Industrial Inspection via UAV

Marcus Chu, Shreyas Sharma, Peter Tong

Phoenix Squadron
Concordia International School Shanghai
999, Mingyue Road
Jinqiao, Shanghai
China, 201206

I. INTRODUCTION (CHANGE TO SOUND LESS LIKE ABSTRACT)

This research paper presents a complete engineering workflow toward a fully autonomous and self-reliant Unmanned Aerial Vehicle (UAV) petrochemical survey system utilizing onboard computing and path planning systems. The first part of the Research Objectives aims to establish the many advantages of such a system, as well as provide a brief overview of technicall objectives and an overview of the system essential for understanding the rest of this article. The second part of the Research Objectives presents our findings on implementing a self-reliant path-planning system on an open-source flight controller and adapting said system to function without any remapping or other assistance required. The third part of the Research Objectives summarizes the engineering workflow required to fuse the gas sensing system with the path planning system, making the drone capable of autonomous inspections. Conclusions then provide a technical bid for such a system, complete with end-user knowledge such as projected operational costs, implementation of a professional workflow, as well as a repository of test materials and supplemental readings.

II. RESEARCH OBJECTIVES (PART ONE)

TODO: GLOSSARY OF TERMS, MERITS (CHANGE TO SOMETHING ELSE), TECHNICAL OBJECTIVES

a. GLOSSARY OF TERMS

1) Pixhawk (PX4): Open-source flight controller for a variety of UAV systems and in wide adoption by many industry leaders including Freefly Systems and Auterion[1]. Used in this system to control low-level flight of the UAV

2) Nvidia Jetson Nano SoC: Small single-board computer with integrated graphics card, capable of computationally intensive tasks, widely used in robotics. Used in this system to control high-level flight of the UAV such as calculating the path of the UAV as well as fuse sensor data. Current one used on Project Prokyon is a “国产” model, meaning that it is a Chinese clone of

3) ROS (Robot Operating System): A open-source collection of libraries and tools that streamlines building robot applications. Interfaces with PX4 and Gazebo/Rviz SITL simulation, for easy application testing[2].

4) MQ (Modular Quantifier): The key core of this project which offers a suite of sensors that can easily be swapped out, with out being reprogrammed. This gives the drone a fully modular and customizable payload option.

5) Intel Realsense D435: Stereo depth camera widely used in robotics autonomy. The camera enables robotic systems to perceive depth in an image based off the laser module built into the camera.

6) Point Cloud: A 3D collection of pixels, often overlaid on an RGB image, that represents points and distances to the sensor, a nd is used to enable drone autonomy as it perceives its environment

b. TECHNICAL OBJECTIVES

I. BUILD OPERATIONALIZED UAV PLATFORM
II. INTEGRATED ON-BOARD PROCESSING COMPUTER
III. INTEGRATED FIRST PERSON VIEW (FPV) VIEW CAMERA AND AUTONOMOUS FLIGHT CAMERA
IV. DEVELOPMENT OF MODULAR PAYLOAD SYSTEMS
V. APPLICATION OF GSS (GAS SENSOR SUITE) SENSOR SUITE
VI. SENSOR FUSION TO ENABLE PATH PLANNING

c. MERITS

THE ADOPTION OF SUCH A SYSTEM WILL ALLOW BUSINESSES TO INCORPORATE A LOW-COST AND MODULAR UAV SYSTEM, CAPABLE OF VARIOUS USE-CASES ACROSS INDUSTRIES, GREATLY REDUCING COST AS A ONE-SIZE-FITS-ALL SOLUTION. AS INDUSTRY 4.0 IS BEING ACHIEVED, AND INDUSTRY 5.0 IS RIGHT AROUND THE CORNER, IT IS CRITICAL THAT WE NOT ONLY ADVANCE THE TECHNOLOGY WITHIN THE MANUFACTURING FACILITIES BUT ALSO THE EQUIPMENT USED TO MAKE SURE THE FACILITY IS RUNNING AT PEAK PERFORMANCE. CURRENTLY, MANY OF THE OFFERED SOLUTIONS ARE EXPENSIVE, AND AN ORGANIZATION WOULD REQUIRE MULTIPLE DRONES IN ORDER TO ACCOMPLISH INSPECTION REQUIREMENTS, AS WELL AS A WORKFORCE, WHO ARE REQUIRED TO PILOT THESE CRAFTS. PROJECT PROKYON BRINGS A NEW SOLUTION TO EXPENSIVE, AND LABOURSOME INSPECTIONS. PROKYON AIMS TO AUTOMATE THE INDUSTRIAL INSPECTION PROCESSES, AS WELL AS ENABLE ONE DRONE TO BE USED FOR VARIOUS TASKS. PROJECT PROKYON UTILIZES A FORWARD-FACING INTEL REALSENSE D345I DEPTH PERCEIVING CAMERA FOR COMPUTER VISION, AND IMAGE RECOGNIZING. THE MODULAR 3PIN STANDARD SENSOR BAY ALLOWS OPERATORS TO EQUIP PROKYON WITH SPECIALIZED AND TASK-SPECIFIC INSPECTION NEEDS EVOLVE. FURTHERMORE, THIS SYSTEM WILL FURTHER IMPROVE EFFICIENCY OVER CURRENTLY AVAILABLE INSPECTION PLATFORMS BY AUTOMATING INSPECTION WORKFLOWS, ALLOWING FOR MINIMAL PREPLANNING. ONE OF THE KEY ASPECTS TO INCREASED EFFICIENCY WITH PROKYON IS AN INBUILT WEB-CRAWLING BOT, THAT ALLOWS THE DRONE TO AUTOMATE THE DATA UPLOAD PROCESS. WITH EXISTING DRONES, OPERATORS MUST MANUALLY REMOVE THE SD CARD, THEN ORGANIZE AND COLLECT THE GENERATED DATASETS. PROKYON REVOLUTIONIZES THIS PROCESS BY AUTOMATING THE UPDATE PROCESS AND USING VARIOUS MODELS THAT CAN BE SWAPPED OUT FOR TASK-SPECIFIC MODELS. HAVING THE ABILITY TO RUN ONE’S OWN MODEL IN ACCORDANCE WITH THE EXISTING FLIGHT SOFTWARE ALLOWS FOR REAL-TIME, AND FAST PROCESSING OF THE IMAGES COLLECTED. ONCE THE DATA HAS BEEN LOCALLY STORED ON THE ONBOARD PROCESSING COMPUTER, THE DATA IS UPLOADED TO AES-256 ENCRYPTED SECURE CLOUD SERVERS, AS WELL AS A VERSATILE BLOCKCHAIN NETWORK – AFTER PROCESSING THE DATA FOR ANY ANOMALIES. WHILE A PLETHORA OF OPTIONS IS AVAILABLE FOR

BUSINESSES TO CONDUCT INDUSTRIAL INSPECTIONS, THESE OPTIONS ARE INCREASINGLY BEING A LESS VIABLE OPTION AS PRICES INCREASE AND INSPECTION NEEDS EVOLVE, NECESSITATING A LOW COST YET MODULAR SYSTEM, CAPABLE OF ADAPTING TO A WIDE VARIETY OF INDUSTRIES. THIS SYSTEM PRESENTS A REVOLUTIONARY INNOVATIVE APPROACH THAT ADDRESSES KEY ISSUES FACING TODAY'S INSPECTION SYSTEMS, AND (HAS) A FRAMEWORK FOR CONTINUED EVOLUTION IN EVER CHANGING ENVIRONMENT.

d. COMPONENTS AND HARDWARE

1) PIXHAWK 2.4.8

The Pixhawk 2.4.8 offers a cost-competitive, unique, and open-sourced flight controller. That offers reliability and stability flight control. The Pixhawk 4 is also compatible with telemetry antennas, as well as options for a lot of expansion. It supports direct communication with the onboard processing computer (Jetson Nano) as well as expandability to incorporate flight sensors.

2) JETSON NANO

The Jetson Nano allows for real-time edge computing of computationally intensive tasks such as path planning, enabling the UAV to operate in GPS-denied environments and without any additional computers or networks

3) MQ GAS SENSORS

The MQ gas sensors offer modular and cost competitive gas sensing solutions, able to interface with the Jetson Nano directly.

4) MN8 GPS:

With the Mn8 GPS, the drone has a redundant flight positioning system, as well as the ability to maintain position, both altitude and attitude.

5) 400KV MOTORS:

Due to the increased distance between the central deck and the motor mounts, more space is available for large diameter (15inch) props. As prop speed increases, there is a diminishing return of speed and thrust generated, and in addition efficiency also decreases. Therefore, limiting the speed of the motors will increase efficiency, making for a more efficient propulsion system and overall, a more power efficient UAV.

6) 40A ESCS:

With 40-amp ESCs, precise control of motors will be enabled, along with advanced motor telemetry and sensor data, allowing the UAV to comprehensively manage all components of the propulsion system

7) ZD850:

The ZD850 offers a large, and expandable Hexacopter drone frame. This gives Phoenix Squadron the ability to lift and carry heavy payloads for a long duration of time.

8) 15" PROPS

9) 10,000MAH BATTERY:

The 10,000mAh battery gives Prokyon the ability to stay airborne for long periods of time, while still giving off 25.2volts of current to power all the on-board electronics and avionics.

10) AT10 II CONTROLLER

The AT 10 II Controller is a local, cost competitive, and reliable alternative to more renowned global brands, such as Taranis controllers.

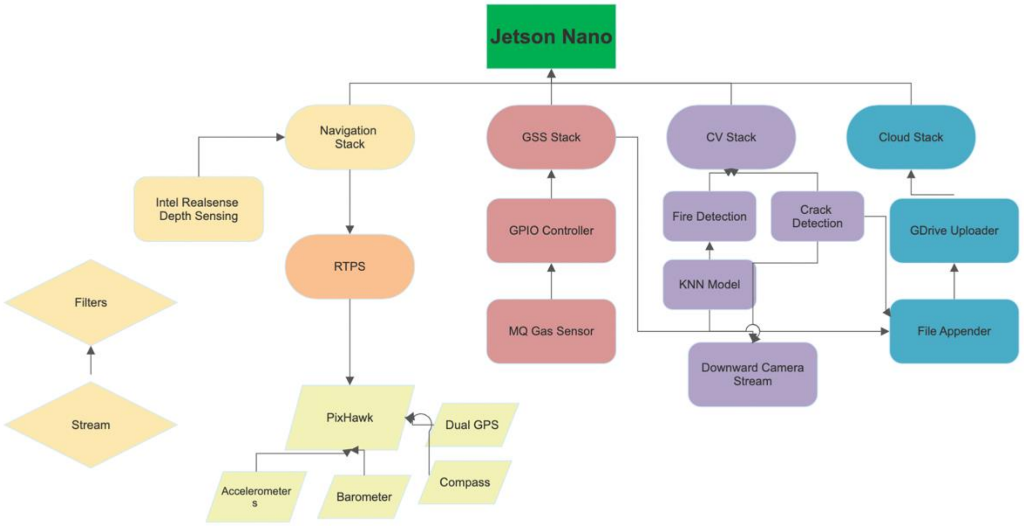
e. SOFTWARE

1) Q-GROUND CONTROL
2) PYCHARM
3) C++
4) PYTHON – PYTHON 3.7 OR 3.8
5) LINUX – UBUNTU 20.04
6) GAZEBO
7) RVIZ

f. Innovating existing solutions

Typical Solution	Project Prokyon
Superior: - Available commercial - Usage of vast GPS systems - Ready to go deployments	Superior: - Autonomy - Modular payload - Cost competitive
A typical solution that is commercially available for these sectors is larger quadcopters that must be manually piloted and are reliant on humans to perceive and analyze the data captured by the onboard camera and customized payloads attached.	Project Prokyon utilizes an advanced stereoscopic sensor suite to utilize SLAM and advanced path planning to enable full autonomy with no preplanning, and therefore true autonomous inspections.

PROJECT FLOW CHART:



III. RESEARCH

In regard to the sensors utilized by Project Prokyon, Project Prokyon utilizes a Intel Realsense depth camera in conjunction with a obstacle avoidance stack to allow true autonomy while in flight. This is combined with A downward camera and GSS (Gas Sensor Suite) to allow the drone to collect data on pipeline anomalies and gas leaks

I) Downward Camera: Image Processing

Within Project Prokyon, there are 2 main scripts and processing algorithms that are running for downward image recognition over the intended structure of interest. Both are based on the Python library OPENCV-PYTHON, commonly referred to as CV2. CV2 also allows for images to be brought in by an external camera, which for Project Prokyon, is being conducted by a downward facing GoPro Hero 4, recording at 480p. The rough FPS received by the system stands at around 20 FPS, limited by the available and allocatable processing power of the Jetson Nano.

First: The Crack detection algorithm (CDA). The algorithm itself works by locating lines with a sharp contour, then within that, checks for height and area. With these data points, the algorithm can itself predict whether there are cracks or not. This algorithm is not a model-based algorithm, but instead purely a computer vision-based system. The CDA displays 3 key frames form the 1 image processed. First: The standard unedited view frame. This frame is needed for verification of cracks and ensuring that the drone is on the right pipeline. This also gives an unedited view for onsite personnel who would review the refined data from Prokyon. Secondly, we have the processed and marked image frame. This is created by taking the standard image feed, then based off the greyscale recognition, via CV2 library boxes, drawn around, and clearly marked pipe cracks. Lastly: The Binary view frame. This itself is a Greyscale camera feed. When the image is converted grey, it allows for extremely efficient computation. As such, variances in the pipe surface are best marked by Binary representations of the image in a Grayscale feed. Anomalies along the pipe surface often produce a quantifiable image phenomenon.

To convert our standard frame into a gray-scale frame, we apply the weighted greyscale image conversion method. The average gray scale conversion method works like such:

$$Grayscale = (R + G + B)/3$$

However, a slight problem lies with using this. This method, simply, does not work as well. Human eyeballs react differently to RGB colors. Eyes are most sensitive to green light, less sensitive to red light, and the least sensitive to blue light. Thus, the three colors should have different weights in the distribution. Insert Ref Based on these findings, we apply the weighted method:

$$Grayscale = 0.299R + 0.587G + 0.114B$$

This method, known as the luminosity method, instead of the simpler direct division used in standard conversion, factors in the respective wavelengths when converting an image grey, which gives a much clearer picture, with true Grayscale colors.

<u>Chronological Execution Example:</u>

1. Camera frame is initialized
2. Video Capture system launches
3. When crack is detected, Binary, Processed and Marked are saved to RAM
4. Upon landing, Crack Detected Images are written to path
5. Path is passed over to the Cloud Stack

The second algorithm that runs on the downward facing camera is fire detection. In Industrial complexes, there is always a risk of fire. However, many of the fires are on a much smaller scale, and quite difficult to detect, especially on longer range gas pipelines. For this, Prokyon also features a specialized model-based algorithm, that fuses data from the MQ2 series smoke sensor, and the custom KNN neural network to find fires, and then mark, and save their images.

II) Intel Realsense: Depth Perception

Obstacle Avoidance Stack Operation Procedure

The Intel Realsense depth camera allows for depth sensing utilizing stereo depth to create a color-coded image of distances to obstacles in the FOV of the camera. As the obstacle avoidance stack operates, there are a few key processes that it must do:

1. Acquire Intel Realsense depth data
2. Filter Intel Realsense depth data
3. Make detections on depth data
4. Interface with Pixhawk and send setpoint messages
5. Inference on progress (planned route) using depth and GPS data

Acquire Data

Figure A: Intel Realsense D435 Depth Camera

The Realsense depth data is acquired by using the pyrealsense2 library in Python, which also generates an RTPS stream which can be viewed by the ground station. It then utilizes a .json file to set parameters of the camera that are most suited. For example, the camera can be tuned to higher speed or higher accuracy, and can also be optimized for environments, such as darker environments.

Filter

This data is then filtered using a spatial edge-preserving filter, temporal filter, and a hole's filling filter.

Spatial Edge-Preserving Filter

The spatial edge-preserving filter allows the Intel Realsense to maintain high definition between edges of obstacles in a complex environment, ensuring that obstacles are measured to the highest efficiency. A transform is utilized for edge-preserving filtering and such images obtained by filtering can be expressed by

$$J(p) = Z\Omega I(q)F(\hat{}p,q\hat{})dq,(1)$$

And the bilateral filter kernel is given by

$$F(\hat{}p,q\hat{}) = G\sigma s(kp-qk)G\sigma r(kI(p)-I(q)k),(2)$$

However, the issue then becomes the topic of preserving distance while also minimizing the error in the edge calculations. Therefore, an isometry, or a distance preserving transform, is used such as a curve of

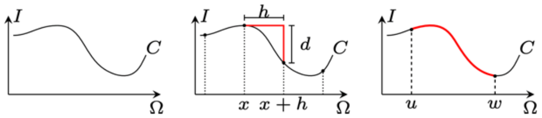


Figure 2: *Curve C defined by the graph* $(x, I(x))$, $x \in \Omega$ (left). *In ℓ_1 norm,* $\|(x+h, I(x+h)) - (x, I(x))\| = h + d = h + |I(x+h) - I(x)|$ (center). *Arc length of C, from u to w* (right).

Therefore, this filter allows for accuracy in detecting obstacles while minimizing the error incurred in a complex environment, allowing the data to be used for obstacle avoidance.

Temporal Filter

As the drone operates in complex environments subject to change, the depth data can have noise introduced into it, causing irregularities in the obstacle avoidance stack's perception of obstacles, causing oscillations as it perceives incorrect sensor data, then correct sensor data. Therefore, a smoother need to be applied to ensure that the drone is precise. This is analogous to a PID loop, where undershot, in this case obstacles not detected should be minimized, and overshot, where obstacles are erroneously detected and therefore causing decreased efficiency. The filter performs a path over the data which will adjust depth values and update tracking history. The filter also accounts for jumps in depth data where there may be a time where there is no depth data. The filter takes the last point of data before the loss of depth data, and the first point of depth data, and extrapolates the average movement of the frames in between to maintain depth data. Therefore, this ensures that the obstacle avoidance stack is always running on data

Holes Filling Filter

Additionally, there may be instances where there are simply holes in the depth detection, which may in fact be small obstacles/gaps in obstacles. Therefore, the system operates on the principle of more inaccurate detections than less correct detections, and therefore will always take the conservative route to an obstacle. Therefore, when there are gaps that are analyzed to possibly be erroneous, the filter will assume and extrapolate depth values from neighboring pixels.

The obstacle avoidance stack then uses this to determine where the obstacle is, and therefore avoid and plan a path around the obstacle. During the execution of the planned route, feedback data is sent to the obstacle avoidance stack which therefore allows it to replan dynamically to match an environmental change.

Detections

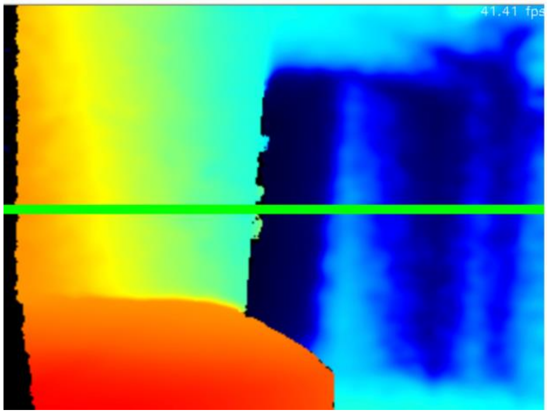


Figure B: Depth Feed from Intel Realsense

Figure B shows the feed from the Intel Realsense. The detections are conducted using a simple algorithm which polls the camera on distances near the green line and builds a 45-degree detection zone. For a forward-facing camera facing in the direction of normal flight for a drone, this means that any obstacle in front of the drone can be detected, but obstacles behind the drone are not detected unless another sensor is placed there. The green line also serves as a filter to assist the drone to distinguish the ground from obstacles, to ensure that the drone does not gain altitude excessively. As obstacles are detected, they are marked in relation to the drone utilizing the depth sensing camera and passed to the planning algorithm to plan a path around obstacles. For example, if a obstacle is detected to the left of the drone, the drone will poll the forward facing and leftwards pixels to determine if those sides are also obstructed. It then plans a path that has the shortest distance to the GPS waypoint, and in real time follows that route. Additionally, if the drone detects an obstacle under it or a large obstacle in front of it, it utilizes behaviors where it will attempt to increase altitude or reverse and find a way around the large obstacle.

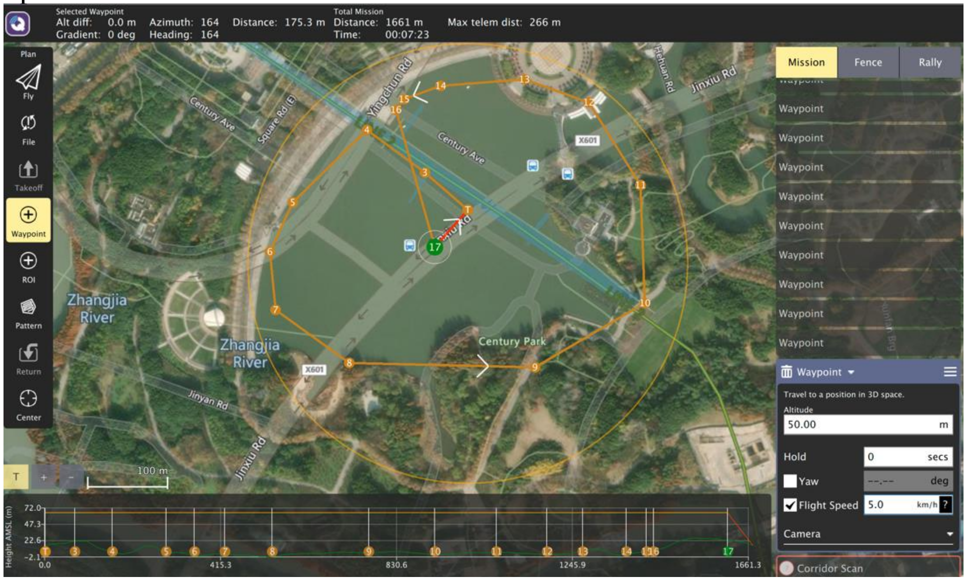
Interface With Pixhawk

Pixhawk is readily configured out of the box to be able to interface with devices utilizing the TELEM2 port on the Pixhawk, with the Pixhawk side running a RTPS client, and the companion computer side running an agent that can communicate with programs through the Mavros library. Additionally, Pixhawk can be configured to accept offboard control, where a companion computer will take over the flight of the aircraft. This is accomplished by the sending of DIST_OBSTACLE messages to the Pixhawk, which will then be able to take over the flight of the aircraft and prevent the pilot/planned route from moving the aircraft in the direction of the obstacle or to find a direction with no obstacles and change the path accordingly. On the hardware side, the TELEM2 port is connected to the Jetson through a USB FTDI adapter and the Pixhawk can then be interfaced to like any USB device, simplifying the process of connecting the 2 devices together. Additionally, the RTPS bridge serves to sync the Pixhawk and Jetson together through a consistent callback function sent through the RTPS bridge, ensuring that calculations are occurring in real time. In addition, the Jetson and Pixhawk are not limited to calculation then running but can recalculate in real time due to the callback function, enabling accurate autonomous flight without packet loss/control inconsistencies, allowing this system to be truly deploy and go, without any other human operation.

III) GPS Way pointing

For efficient operation of a drone inspection system, operators need to be able to quickly assign drones to survey paths using commonly available methods. While manually programming flight routes allows operators to precisely determine which areas need inspection and the route to take, it requires a lot of time and is inefficient for every route to be planned and even replanned if unexpected obstacles are present, hindering operations. Therefore, Project Prokyon utilizes GPS way pointing but only in a supplemental sense. As GPS waypoints are transferred to the drone through the ground control software, the drone plans a straight-line path between the points, as the environment is assumed to be free of obstacles, and therefore these GPS waypoints are easy to plan, as operators can reference satellite maps. However, these paths do not account for obstacles. Therefore, the obstacle avoidance function is introduced to be a "check" of the GPS planned route. The obstacle avoidance function is capable of overriding the GPS planned route to avoid obstacles, all while maintaining path towards intended points & thus, allowing for inspection to continue its own.

GPS waypointing works on the principle of longitude & latitude. Each point that is marked on the flight path will have its own specialized coordinates that are specific to that area. In addition to GPS coordinates, we can set some important parameters.


Selected Waypoint
Alt diff: 0.0 m
Gradient: 0 deg
Azimuth: 164
Heading: 164
Distance: 175.3 m
Total Mission
Distance: 1661 m
Time: 00:07:23
Max telem dist: 266 m

Travel to a position in 3D space.
Altitude 50.00 m
Hold 0 secs
☐ Yaw --.-- deg
☑ Flight Speed 5.0 km/h

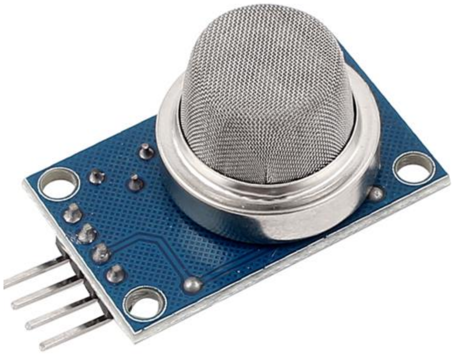
Figure C: QGroundControl GPS Waypoint Interface

As seen above in the figure, there are 16 key waypoints we want the drone to follow. Each one of these waypoints has a specific altitude that we want the drone to meet. Outside, we can see a larger circle, which is a much larger fence. This fence acts as a geofence and informs the drone that it cannot cross this area. Up top, we can see some flight information, such as time, flight distance (Scalar), and the max telemetry distance, which is critical for making sure the drone remains in the range all the time.

As the path is being executed, if any obstacles come in the way of the direct flight plan, then the depth perception camera algorithm will plan a new flight plan around the obstacle, making sure its distance from the original waypoint is as less as possible.

IV) GSS

Apart from detecting anomalies in pipes, it is also important to enable the drone to detect leaks of petrochemicals as leaks detected are a certain indicator of a pipe anomaly that may not be detected even by human inspectors carrying sensors. Therefore, the drone is equipped with 6 different gas sensors. These sensors utilize the downwash that the drone propellers to feed in data from the sensors and the GPS coordinates are fused together to provide a full and functional reading of the GSS sensor suite to pipeline management personnel. The GSS script is written in such a way that it is compatible with more than 2000 sensors out of the box that feed number-based sensor values via the digital pins on the Jetson Nano. It is also possible to use the analog and GPIO pins.



.

Figure D: MQ Gas Sensor

For Project Prokyon, we have selected to use MQ series sensors[3] on the GSS (Figure C). These sensors are low-cost and ultra-efficient. Essentially, they work on the simple principle of checking the resistance of the sensing material inside. The sensing material is an Aluminum Oxide based ceramic, which is coated with Tin dioxide, and entirely enclosed in the stainless-steel mesh in the core of the sensor. The sensing element has six connecting legs attached to it. Two leads are responsible for heating the sensing element, the other four are used for output Pulse width modulation (PWM) signals. Oxygen gets adsorbed on the sensing material when it is heated in air at a high temperature. Then donor electrons present in tin oxide are attracted towards this oxygen, ultimately preventing current from flowing through the sensing material. When reducing gases are present, these oxygen atoms react with the reducing gases decreasing the surface density of the adsorbed oxygen. Thus, the sensing material now can conduct electricity. As electricity is conducted through the sensing material, the voltage values are measured to predict the concentration of gas, thus giving the values of gas present. These voltage values are then forwarded to any interfacing controller, which in our case, is the GPIO of the Jetson Nano.

The advantage of having such a system like the GSS is that of expansion, cost competitiveness, and versatility. Previously, most pipeline inspection drones would not carry any sensors, but instead would be solely reliant on their camera to detect and recognize cracks. While that is often accurate, there is always a possibility that the camera has not seen something. In any case, the previous usage of sensors would be hand-held sensors, which would manually be carried around by pipeline management personnel – which would not only prove to be an expensive and inefficient task, but also a herculean one.

As the drone is in flight, the GSS script logs all sensor values to a .csv file, where it is then processed by the algorithm to determine if any values are beyond the nominal range and proceeds to pass the file onto the cloud stack.

The GSS can also be replaced with a different sensor, including handheld sensors that can interface via USB to the Jetson Nano.

V) Cloud Stack

The Cloud Stack, also called the Cloud API, is the final step in data for Project Prokyon. Effectively, in a standard run of the project, all the data, including the GSS .csv log, and the downward facing camera, as well as the real-time route followed by GPS coordinates are all logged into the RAM. Once they have been logged onto the SD card of the Jetson Nano, they are then appended by the Cloud stack's set directory essentially, a web-crawling bot copies and pasts all the files into a cloud server of our choice, whether it be Google Drive, or a custom localhost server running on the same network as the Jetson Nano.

The cloud stack is heavily reliant on the OAuth process. Google itself does restrict bot activity, and the process of bots sending too much traffic. These bots are detected via captchas and RE-captchas. OAuth is an open standard for access delegation, commonly used as a way to access password-protected websites but without giving them the passwords. We opt to not send any passwords because we access HTTP, not HTTPS. What this means is anyone can intercept the information, and passwords and edit the data before it is pointed into the cloud source. To protect the file from being cracked or intercepted during transmissions via man-in-the-middle (MDM) attacks, the entire .csv file is AES256 encrypted for safe and secure transmissions from Jetson to server. While the cloud stack has primarily been developed for Google Drive, which is a cloud-based process, it has support for integration into a blockchain based data storage system, such as Graphite. For local host solutions, it is also possible to forward the port and IP address to the selected ones.

Many of the cloud hosting servers will require a virtual-proxy-network (VPN) if they are being implemented in China.

IV. RESULTS

Overall Research Findings

It is found that such an autonomous drone system with full autonomy capable of autonomous inspections through the use of computer vision and chemical sensors is capable of being implemented as a cost competitive and efficient solution to petrochemical inspections. Our system accomplishes obstacle avoidance with 90% of obstacles successfully avoided, 90% detection accuracy of pipeline damage, and 98% chemical sensing accuracy. Furthermore, its cost competitiveness is heightened by the use of COTS (Commercial Off the Shelf) components such as a Nvidia edge computing computer and modular drop-in replacement gas sensors. Furthermore, for a company seeking such a solution for petrochemical inspections, the introduction of the system we propose will be streamlined as no additional resources/arrangements are needed for such a system, as the system only requires GPS waypoints, on which current semi-autonomous drone systems operate.

Functionality-Specific Research Findings

I. Obstacle Avoidance System Test Method

Regarding obstacle avoidance for ground platforms, such systems exist and are in sufficiently widespread use to allow for integration on ground robotics platforms. However, in the field of aerial robotics, much progress is being made, but such obstacle avoidance/path planning systems are not as widespread as ground robotics obstacle avoidance/path planning systems. Furthermore, our path planning algorithm does not require high computational power, as only obstacle distance data is computed. While the definite fastest approach is planning a path before the obstacle, so that the drone is not in close range to the obstacle, we instead utilize a slower, but much more computationally efficient method. Our method utilizes the two targets that the drone must obey, the GPS waypoint, where it will always fly the shortest route to the waypoint, and the obstacle avoidance where the drone will not move forward. Therefore, the drone is able to conduct path planning in real-time with a low computational requirement. Furthermore, the callback function that is running between the Jetson and Pixhawk is shown to be a effective and precise way of drone control, shown by the following graphs, the first of which demonstrates a sub-meter position accuracy of the path planning algorithm, and the second of which shows the low computational requirement of the path planning algorithm.

Figure E: Altitude Estimate (Setpoint compared to GPS Altitude)

In Figure E, where the Altitude Setpoint is the commanded altitude by the Jetson, and where GPS Altitude is the altitude of the drone when that setpoint was set, there is almost no deviation, with the largest deviation being 1 meter. This shows that the callback function is able to accurately control the altitude of the UAV based on the path planning algorithm to meter accuracy, allowing it to operate in complex environments.

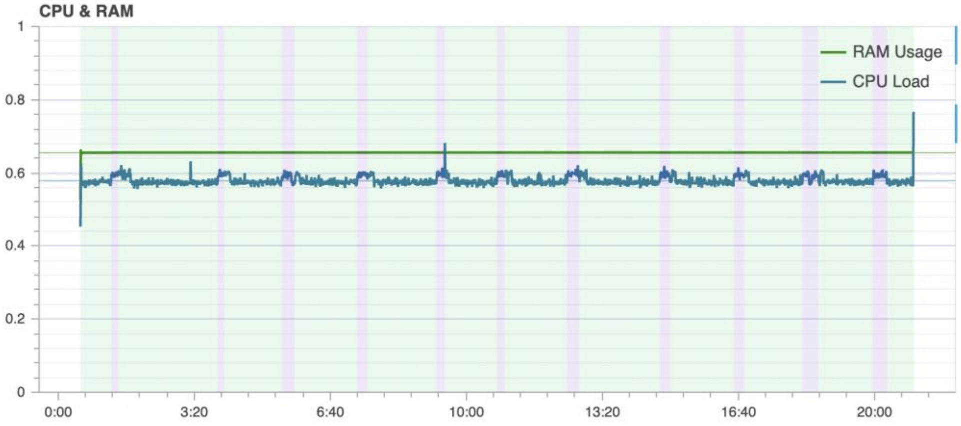


Figure F: CPU and RAM usage

In Figure F, the CPU and RAM usage is observed to be below 70%, thus showing that the path planning algorithm is computationally efficient and does not require a large amount of computing power

II. GPS Waypointing

GPS waypoints are quite an effective, and accurate way of controlling and sending points to the drone. The average deviation from the set waypoint is less than 47 centimeters. A key problem of GPS waypointing is its reliance on an external, and uncontrollable variable: GPS Satellites. These are not always stable, as at certain times of day, the drone itself can only connect with a certain number of satellites. While this is an efficient control method, it does certainly come with drawbacks, making it only reliable in certain locations with good GPS, and cellular connectivity. In a petrochemical inspection scenario, most of the inspected areas would be open, therefore GPS waypointing is found as an effective solution to marking areas of inspection for the drone.

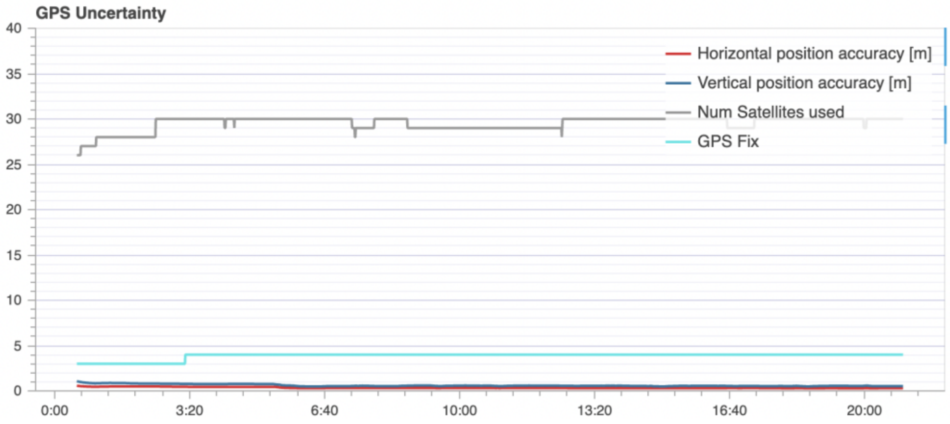


Figure E: GPS Uncertainty (Horizontal position accuracy and Vertical position accuracy compared to Number of satellites used)

III. GSS

While most inspection methods make use of expensive sensors often used to detect just one type of chemical, our system utilizes modular COTS sensors. While these sensors can detect chemicals extremely well and efficiently, for such an aerial inspection system to utilize these sensors would make it infeasible to use widespread. Therefore, our system makes use of commonly used gas sensors with the added advancement of downwash sample collecting or utilizing the downwash of the drone to force air collecting samples into the sensors. Therefore, the drawbacks of using such COTS sensors including low range are mitigated, allowing for the use of low-cost, modular, and therefore easily used sensors. Therefore, our system brings the benefits of modular and COTS sensors while allowing for accurate and efficient data recording. Ultimately, the GSS, while holding drawbacks with sensitivity & range, proved to be an excellent development platform for early testing & prototyping.

IV. Cloud Stack

Our cloud stack simplifies collecting data from our system by not necessitating a separate system, but instead by integrating into existing data management systems. Through the use of the Cloud Stack, data can be uploaded to any data management system utilizing csv format, greatly simplifying the integration of the data from our system. Furthermore, the use of blockchain data transfer enables our system to be a secure system capable of meeting industry standards. The cloud stack and API worked superbly when connected to a mobile hotspot. A future expansion upon this would be adding a 5G LTE network module and card adaptor to the Jetson, to allow simultaneous uploading and processing of data accumulated over the flight duration.

V. Crack Detection

The utilization of computer vision to detect anomalies in pipes is not only a feasible solution to implement for a wide variety of inspections, but also one that remains relevant. Our computer vision algorithm achieves 90% accuracy in the detection of pipeline damage with the use of COTS equipment, further allowing it to be used and more widely adopted due to low cost and availability. Furthermore, unlike traditional sensors which only detect one or a few gases, and are only applicable to these gases, our crack detection algorithm only requires that the detected anomaly is visible, and therefore the algorithm is applicable to many more situations requiring detection of anomalies. However, with COTS equipment compared to functionality-specific sensors, less suitability for a specific use case is present. In the case of the crack detection system, the imaging system used, Hero 4, does not produce as clear an image, especially at altitude. Therefore, with the addition of a zoom camera, the crack detection system would be able to better detect anomalies.

Future Advancements and Outlook on Project Prokyon:

While we have a fully functional, and working prototype, there is room for advancements to increase the accuracy of Project Prokyon. Key advancements on the drone itself would be adding real-time kinematics (RTK) to further boost the accuracy of the drone by giving it a base station to triangulate with. Additionally, Aruco code-based images for points of interest along the pipeline are a feasible and a direct drop-in replacement widely used in other inspection robotics systems to allow for even more precision in inspected areas or inspected equipment, such as individual gauges, further advancing the data that Prokyon can collect, making it an even more versatile platform.

Eventually, Prokyon will evolve into a fully functional swarm of robots, capable of full-scale industrial inspections, and no longer just be limited to pipelines. Working with smaller microdrones, larger hexacopters, ground-based robots, as well as quadrupedal systems, and IOT-based sensors placed within structures, Prokyon will evolve into the all-in-one solution for industrial inspections across a vast spectrum of industries. For more information, References provides links for further reading into the systems and topics utilized in Project Prokyon, as well as a link to the code repository for Project Prokyon.

REFERENCES

[1] https://pixhawk.org/
[2] https://www.ros.org/

[3] **https://www.elprocus.com/an-introduction-to-mq2-gas-sensor/#:~:text=MQ2%20gas%20sensor%20is%20an,in%20contact%20with%20the%20gas**
[4] https://www.dynamsoft.com/blog/insights/image-processing/image-processing-101-color-space-conversion/ - Gray Scale
[5] https://github.com/PhoenixSquadron/Project-Prokyon
[6]

AUTHORS

Shreyas Sharma – shreyas2025165@concordiashanghai.org
Marcus Chu – marcus2025180@concordiashanghai.org
Peter Tong – peter.tong@concordiashanghai.org
