

# **Design and Test of Emergency Response Drifter with Low Debris Profile**

University of Washington

School of Oceanography

Senior Thesis 2017

Charles Brennan

Charles Brennan

(Advisor: Miles Logsdon)

School of Oceanography

University of Washington

1503 NE Boat Street

Seattle, WA 98105

[ccb43@uw.edu](mailto:ccb43@uw.edu)

**Acknowledgements:**

Thank you to Miles Logsdon for his guidance and mentorship over the past year. Without his help, my project would not have been as successful as it was and I would not have learned as many skills along the way. Thank you to Arthur Nowell for his constant support throughout the year. Thank you to the fellow ocean technology students for their ideas, care, and support that helped me push through the stressful times when things did not go my way. Thank you to Julianna Chen for her help with coding for the GPS.

**Abstract:**

Affordable techniques for tracking point source water pollutants such as small-scale oil spills are limited to in situ drifters and manually controlled drones. The new age technology of low cost, areal footage via an off-the-shelf drone provides benefits of visual, real-time data as well as versatility in its movements and functions. On the other hand, the older technology provides benefits of extremely low cost per drifter and the ability for them to stay on station for days at a time providing data. Both platforms have negatives whether that is comparable cost, battery life, manpower or versatility. These negative aspects of either technology are exposed through this experiment. The question that is answered by this report is which technique for tracking point source pollutants is more effective? The method for testing this question involves building and employing both systems and comparing the precision and accuracy as well as weighing the positives and negatives. After conducting these tests, the older technology of deploying drifters is a more affordable, accurate, and precise method for tracking a spill. With improvements—and a lot more money—both techniques have the potential to be more effective. The simplicity of the drifter remains the reliable, efficient method due to its low cost for a high number of units and its on time on station abilities.

**Introduction:**

According to the US National Research Council (NRC, 2003), roughly one-third to one-half of the oil discharged into the sea is from natural seeps, with the rest coming from anthropogenic sources. These types of discharges can be devastating to wetlands, coral reefs, and fishing grounds as well as to bird, fish, shellfish, and marine mammal populations (Christopher and Apel 2004). This project focuses on point source spills that can be either from natural causes or anthropogenic sources. In oil spills such as the Deepwater Horizon oil spill in 2010 ocean circulation is fundamental to planning mitigation strategies and to determining both landfall of oil and movement of oil toward biologically sensitive areas in deep and shallow waters (Yonggang et al. 2011).

Both currents and wind move oil spills on the surface of the ocean. The velocity of a spill is the vector sum of the surface ocean current at that location plus an empirical wind-induced drift. The magnitude of the wind-induced drift is 3.5 percent of the local wind speed (Price et al. 2003). In a case where the wind speed is 10 knots and the surface current is two knots a 3.5 percent wind-induced drift would cause the surface pollutant to have a velocity of 2.35 knots. Although the wind does have some effect on the drift of the surface pollutant, the majority of the force is from the surface currents. Because of this, the most efficient way to track a point source pollutant such as oil is to track the surface expression of that pollutant.

Low impact, affordable, and easily deployable equipment is necessary to form an efficient and responsible reaction plan to near shore pollutant spills.

Current methods for tracking oil spills focus on observations made by remote

devices and models of proposed movements. The remote sensing observations consists of using planes or drones, satellite data, and drifters. The importance and efficiency of the use of drifters has been overlooked for more advanced technologies. The main disadvantage of these advanced methods are the cost and time they take to employ. Satellites are extremely expensive to deploy and unless they happen to be over the correct location, it could take too long to acquire the necessary data. Planes also take planning and can be expensive to use for long periods of time in a continuous tracking situation. Although not as expensive as these other approaches, model based tracking can be too inaccurate in a point source pollutant spill situation to be of much use to a cleanup and recovery crew. Drones are a popular alternative to planes and satellites because of there efficiency on small-scale responses. What is needed is an inexpensive, compact, and easily deployable drifter that can be deployed on a moments notice and collect continuous data for days at a time.

## **Design Principles and Implementation:**

For the design of my drifters, I had some simple parameters that needed to be met. First, the drifters had to float on the surface so they could track currents. Second, they needed to remain stationary in the current and not spin or allow the current to simply pass by or through the devices. Third, they had to contain some sort of tracking device to record their movements. Fourth, they also needed a way to fold up and store efficiently as well as to be quickly deployed by one person on a small craft. And lastly, they needed to have a waterproof housing to protect any electronics that would need to be on board. One of the initial ideas was to make these drifters biodegradable to minimize their environmental impact in the case that they are lost. Also, having them transmit data over wifi to a boat or shore would increase their success for real time data acquisition that would aid recovery and cleanup success. Meeting both of these parameters would have pushed the project over budget and therefore were not used in the final design. The final design focused on inexpensive materials that would be lightweight, easy to store, while still providing accurate data on water movement.

## **Testing methods:**

For the testing of my instruments, I started in phases with a testing plan to control and organize the progress of my build. There were two distinct pathways for completion of this project. First, it was necessary to produce code and an Arduino GPS shield setup that could track and store latitude and longitude onto a micro SD card using a Sparkfun Redboard. Meanwhile, the second branch of the testing

process was to work on was the build of the drifter itself. Due to the uniqueness, importance, and timing of both of these parts of the build process, concurrent work was done on these pathways. Long wait times for the specialized PVC parts pushed back the timeline for the case build. This provided added time to work on the code early on in the process although it was not originally planned. For each pathway there were certain benchmarks needed to be hit during the build and testing plan. For the GPS, the necessary materials were attained including boards (Sparkfun Redboard with Arduino), an SD card shield (Sparkfun), and the actual GPS from Sparkfun (EM-506). Then, a simple GPS with code that would write to the serial monitor on my computer was assembled using online sources from the Sparfun website. Then, the challenge began in working to make the code write to a micro SD card on a micro SD shield in the format that was necessary for what was needed. This was tested and put through multiple iterations and was eventually successful. While working on the GPS, gathering materials for and building the main frame and housing of the drifter occurred. Once the necessary materials were obtained, the 3-inch PVC T-joints were configured and tested for how waterproof they would be. They were successful in being the housing for my GPS. Building then continued to configure one drifter in full based off the original drawings and was successful. Once one drifter was fully configured, it was able to be thrown off the dock to test its floating ability and further test its waterproofing. This was the first wet test and it was performed successfully on 03 March 2017. From there, two more fully functioning drifters were built. Then, a complete wet test with a drone was able to be completed on 16 April 2017 to produce the necessary data to make a decision on

which technique was more efficient and accurate. Following this test, the data necessary to comment on the advantages and disadvantages of my design was available.

For the deployment of these devices, each drifter was placed in the bay just south of the University of Washington school of Oceanography building dock. This was to imitate the deployment of these devices on the leading edge of a spill. At each given time, the drifter was observed and their locations were recorded using the camera and onboard GPS of the DJI Phantom 3 Pro drone by flying over each drone at approximately 2 meters and recording their GPS location that displayed on the screen. Once they were logged, the drone rose to the height where all three drones were visible in the frame and the time and GPS location were logged. This was repeated approximately every 5 minutes for ten repetitions. Meanwhile, the onboard GPS was logging the locations of each of the drifters in three-second intervals.

To process this data and make it visually pleasing for the viewer, the data was uploaded from drone recordings as well as from the drifter SD card to ArcGIS. A shape file was created for each drifter with the GPS points acquired by the drone to compare to the GPS data points from the drifter. A layer was made to simplify the shoreline around the University of Washington School of Oceanography dock. Accepting the drifter to be the most accurate GPS points, error calculations were made based off the average distance from the accepted location to the drone recorded location. Sinuosity calculations were then made from the drone GPS data to determine shape change of the front in a quantitative process. Taking the distance

from the first drifter to the second drifter and adding that to the distance from the second drifter to the third drifter measured sinuosity. This value was divided by the distance from the first drifter to the third drifter to calculate the sinuosity. The mapping capabilities in ArcGIS (ArcMAP) were then used to create visual displays, which are shown in figures 2, 3 and 4.

### **Results:**

In the calibration test run by Malea Saul, the mean error by the drone when compared to a handheld reference GPS was 0.7282 meters with an average minimum of 0.113 meters and average maximum of 2.276 meters when flying in a straight path. The average orthogonal distance in meters from the drifter GPS to the drone GPS over ten runs was 6.158 meters with a minimum distance of 1.98 meters and a maximum distance of 11.38 meters.

### **Discussion/Conclusion:**

The accuracy test of the Phantom 3 Pro's inflight GPS revealed a lack of both accuracy and precision. When comparing the Phantom's GPS average error against a standard handheld GPS, the drone mounted GPS was not accurate enough for the tested scales. Although there was little difference in error of the drone GPS coordinates when flying at different heights, the coordinates were also not precise. Because the extent of the error for the drone mounted GPS, the recorded GPS coordinates are selected in an area separate from where the drone actually is located. The recorded video confirms that even when the drone is flying in a straight

track over the flags in the ground, the recorded points from the GPS show a scattered pattern moving in the correct direction.

When analyzing the results from the test of the drifter, the inner (closest to shore) edge of the track for the spill as recorded by the drifter GPS is compared to the same track but recorded with the drone mounted GPS. The orthogonal distance from the control—GPS on drifter—to the drone recorded coordinate confirmed the previous accuracy and precision test. During this test the distances and time scales were slightly longer and therefore allowed the visual representation of the drone path to look more accurate. In reality, the data shows that the drone is still inaccurate with an average error of 6.16 m.

The benefits of using the drone for point source pollutant are that there can be very quick response time in terms of being able to process data. A live updating camera feed can be very helpful in spotting and narrowing in on certain areas to view the full extent of spills. This visual aspect can be of help to the eye due to it being video footage. Although the footage is a seemingly helpful feature, the lack of numerical data will hurt response planning for the cleanup process. It takes a considerable amount of time to pull the SD card from the drone and access the metadata to acquire GPS coordinates. Not only is the drone hard to get real-time numerical data off of, but also the unit costs can be very high. Because the purpose of this test includes working over water, it is necessary to have a high quality drone that will not be at risk of crashing into the water. Weather dependency is another issue with the drone tracking. If the wind is too strong or there is heavy rain it will be very challenging to control the drone to the point where it may not even be

possible to fly. To achieve this high quality of a drone along with having an onboard, gimbal mounted camera and enough thrust to fly in the wind, it should be expected to spend no less than \$1000. To add to the expense of the drone, a certified drone pilot will need to be trained and hired to utilize the full capabilities of the device. Lastly, the battery life is a limiting factor when trying to reach out to further distances and still be able to stay on station to collect data. The time wasted changing batteries back at the home base (ship or land) could be crucial lost time for gathering a response to the problem.

The drifter benefits from a relatively accurate onboard GPS most likely due to its much more consistent velocities than the drone. The drifter is also much less expensive per device. Coming in at under \$100 per device, ten could be built for every drone brought. Because the drifters are made to be such a manageable size, one person can deploy them from a small craft or dock and many can easily be placed consecutively along the outer edge of a spill. The most important benefit of using the drifter technology is its strengths in an intertidal zone where data over 24 hours is necessary to analyze the full picture of the situation. The drifter is able to stay on station and provide in situ data throughout the 24-hour period of a tidal cycle. Comparatively, the drone is lacking in this aspect given the drone's flight time of thirty minutes per battery and necessity for a pilot. The drifter will provide in-situ data and does not need to be observed or checked on for the duration of its deployment. The drawbacks of the drifter are that you need a boat to deploy them (unless spill is from the shore or a dock). When storing the units in bulk aboard a ship, they will take up much more space than a drone would. The most important

drawback, and something that should be improved on future generations for this project, is the data communication. The lack of a real time data streaming option limits the ability of the drifter to provide the necessary data for a time sensitive cleanup.

The front edge sinuosity calculations showed how the drifters could provide data on a spill and its progression over time. After time 2, the sinuosity calculations did not change very much over time. This observation from the data illustrates a consistent movement by the oil spill similar what can be seen from the air by a drone-mounted camera. This supports the hypothesis that the drifters properly track an oil spill.

The drifter is a better choice for tracking near-shore point source pollutant spills with the implementation of wireless real time data transfer. The cost benefit analysis favors drifter based off their simplicity, ease of deployment, and low cost of development and production.

**Figures/Tables:**

height (m)	avg (m)	min (m)	max (m)	mean error
3	2.6682	0.411	4.559	
6	3.4666	0.411	7.459	
10	3.0516	0.42	5.856	
Reference	3	avg 0.113	avg 2.276	0.7282

Table 1. Drone to reference path error

Tx_inner edge	orthogonal distance (m)	Average distance (m)	minimum distance (m)	maximum distance (m)
1	3.57	6.158333333	1.98	11.38
2	4.56			
3	9.74			
4	11.38			
5	1.98			
6	5.72			
7	N/A			

Table 2. Orthogonal distance (m) from Drifter GPS track (control) to drone GPS track at time Tx.



Figure 1. Drifter floating in water picture at first wet test.

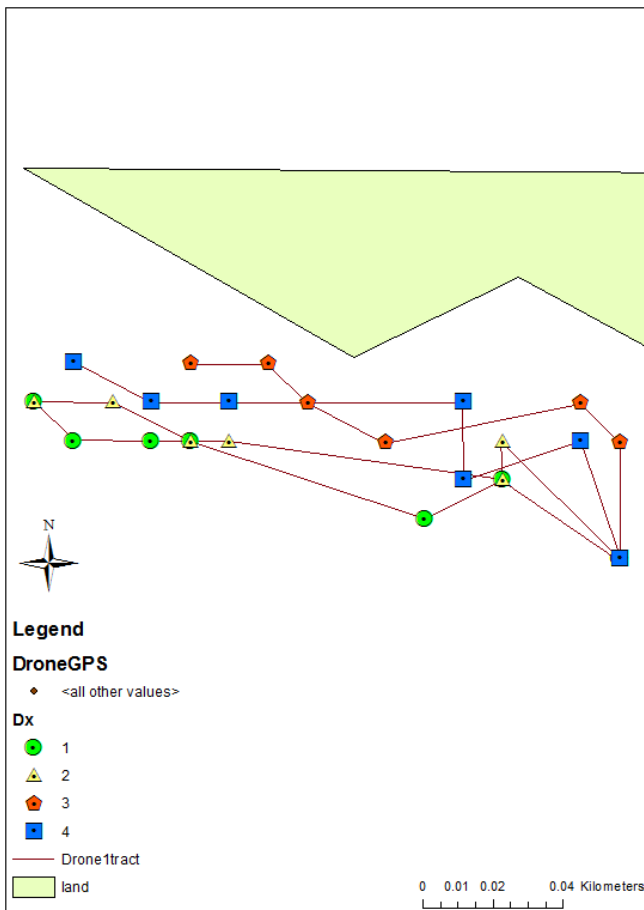


Figure 2. Drone GPS tracts for drifters 1-3 and overall views.

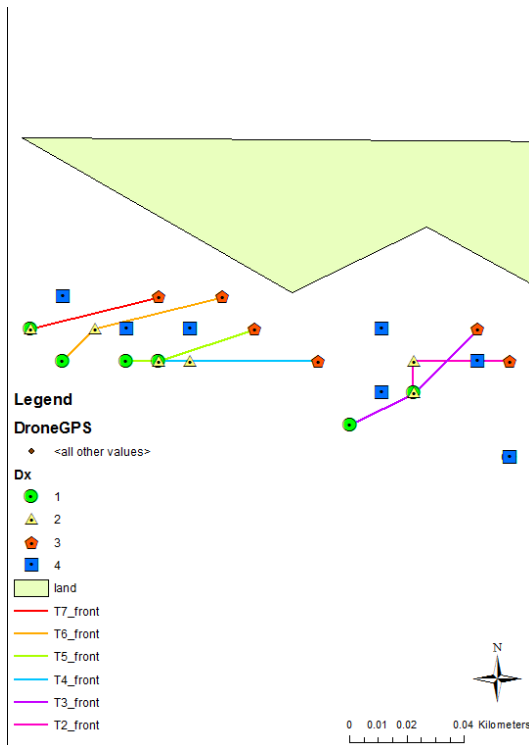


Figure 3. Front lines of sinuosity calculations.

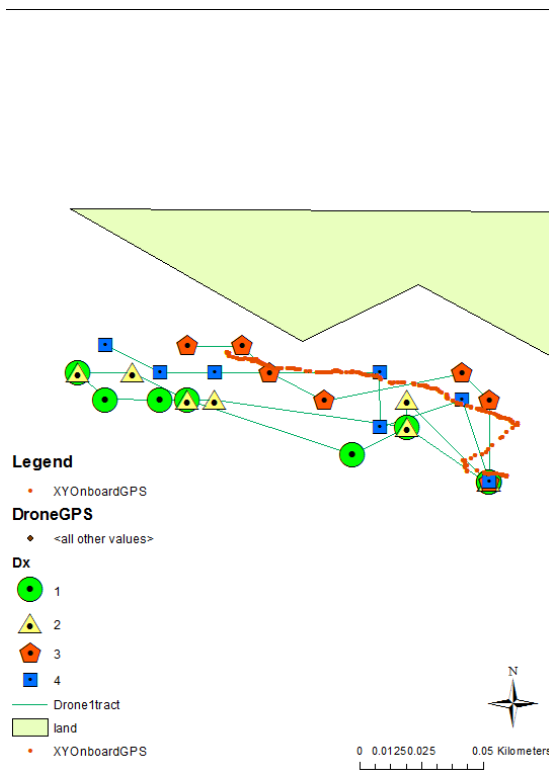


Figure 4. Onboard drifter GPS compared to drone GPS

## References:

Caruso, M.J., M. Migliaccio, J.T. Hargrove, O. Garcia-Pineda, and H.C. Graber. 2013. "Oil spills and slicks imaged by synthetic aperture radar". *Oceanography* 26(2):112–123, <http://dx.doi.org/10.5670/oceanog.2013.34>.

J. M. Price *et al.*, "Evaluation of an oil spill trajectory model using satellite-tracked, oil-spill-simulating drifters," *OCEANS 2003. Proceedings*, San Diego, CA, USA, 2003, pp. 1303-1311 Vol.3.  
doi: 10.1109/OCEANS.2003.178046

R.J. Christopher, J.R. Apel (Eds.). "Synthetic Aperture Radar Marine User's Manual." US Department of Commerce, Washington, DC (2004), pp. 263–276

Yonggang Liu, Robert H Weisberg, Chuanmin Hu and Lianyuan Zheng.  
"Tracking the Deepwater Horizon oil spill: A modeling perspective" *Eos Trans. AGU* Vol. 92 Iss. 6 (2011)

## Appendices:

Code:

```
/*
This example uses software serial and the TinyGPS++ library by Mikal Hart
Based on TinyGPSPlus/DeviceExample.ino by Mikal Hart
Modified by acavis and 2016 Winter Ocean Sensing Class
*/

// import libraries
#include <TinyGPS++.h>
#include <SoftwareSerial.h>
#include <SPI.h>
#include <SD.h>

//create file
File myFile;

//file name
#define fileName "gpsTwo.txt"

// Choose two Arduino pins to use for software serial
// The GPS Shield uses D2 and D3 by default when in DLINE mode
int RXPin = 2;
int TXPin = 3;

// The Skytaq EM-506 GPS module included in the GPS Shield Kit
// uses 4800 baud by default
int GPSBaud = 4800;

// Create a TinyGPS++ object called "gps"
TinyGPSPlus gps;

// Create a software serial port called "gpsSerial"
SoftwareSerial gpsSerial(RXPin, TXPin);

void setup() {
  // Start the Arduino hardware serial port at 9600 baud
  Serial.begin(9600);
  while (!Serial) {
    ;
  }

  // Start the software serial port at the GPS's default baud
  gpsSerial.begin(GPSBaud);
  Serial.println(F("DeviceExample.ino"));
  Serial.println(F("A simple demonstration of TinyGPS++
with an attached GPS module"));
  Serial.print(F("Testing TinyGPS++ library v. "));
  Serial.println(TinyGPSPlus::libraryVersion());
  Serial.println(F("originally by Mikal Hart and adapted
by the 2016 Winter Ocean Sensing Class"));
  Serial.println();

  // Start initialization of SD card
  Serial.print("Initializing SD card...");
  if (!SD.begin(4)) {
    Serial.println("initialization failed!");
    return;
  }
  Serial.println("initialization done.");
  Serial.println();
}

void loop() {
  // This sketch displays information every time a new sentence
  // is correctly encoded.
  while (gpsSerial.available() > 0) {
```

```

    if (gps.encode(gpsSerial.read())) {
        displayInfo();
        printInfo();
        delay(1000);
    }
}
// If 5000 milliseconds pass and there are no characters coming in
// over the software serial port, show a "No GPS detected" error
if (millis() > 5000 && gps.charsProcessed() < 10) {
    Serial.println(F("No GPS detected"));
}
}

```

// method to display the information on the serial monitor

```

void displayInfo() {
    // get and print location
    Serial.print(F("Location: "));
    if (gps.location.isValid()) {
        Serial.print(gps.location.lng(), 6);
        Serial.print(F(", "));
        Serial.print(gps.location.lat(), 6);
    } else {
        Serial.print(F("INVALID"));
    }

    // get and print altitude in meters
    Serial.print(F(" Altitude(m): "));
    if (gps.altitude.isValid()) {
        Serial.print(gps.altitude.meters(), 2);
    } else {
        Serial.print(F("INVALID"));
    }

    // get and print date
    Serial.print(F(" Date/Time: "));
    if (gps.date.isValid()) {
        Serial.print(gps.date.month());
        Serial.print(F("/"));
        Serial.print(gps.date.day());
        Serial.print(F("/"));
        Serial.print(gps.date.year());
    } else {
        Serial.print(F("INVALID"));
    }

    // get and print time
    Serial.print(F(" "));
    if (gps.time.isValid()) {
        if (gps.time.hour() < 10) {
            Serial.print(F("0"));
        }
        Serial.print(gps.time.hour());
        Serial.print(F(":"));
        if (gps.time.minute() < 10) {
            Serial.print(F("0"));
        }
        Serial.print(gps.time.minute());
        Serial.print(F(":"));
        if (gps.time.second() < 10) {
            Serial.print(F("0"));
        }
        Serial.print(gps.time.second());
        Serial.print(F("."));
        if (gps.time.centisecond() < 10) {
            Serial.print(F("0"));
        }
        Serial.print(gps.time.centisecond());
    } else {
        Serial.print(F("INVALID"));
    }
}

```

```

    }

    Serial.println();
}

void printInfo() {
    // open file
    File myFile = SD.open(fileName, FILE_WRITE);

    // get and print location in file
    myFile.print(F("Location: "));
    if (gps.location.isValid()) {
        myFile.print(gps.location.lng(), 6);
        myFile.print(F(", "));
        myFile.print(gps.location.lat(), 6);
    } else {
        myFile.print(F("INVALID"));
    }

    // get and print altitude in meters in file
    myFile.print(F(" Altitude(m): "));
    if (gps.altitude.isValid()) {
        myFile.print(gps.altitude.meters(), 2);
    } else {
        myFile.print(F("INVALID"));
    }

    // get and print date in file
    myFile.print(F(" Date/Time: "));
    if (gps.date.isValid()) {
        myFile.print(gps.date.month());
        myFile.print(F("/"));
        myFile.print(gps.date.day());
        myFile.print(F("/"));
        myFile.print(gps.date.year());
    } else {
        myFile.print(F("INVALID"));
    }

    // get and print time in file
    myFile.print(F(" "));
    if (gps.time.isValid()) {
        if (gps.time.hour() < 10) {
            myFile.print(F("0"));
        }
        myFile.print(gps.time.hour());
        myFile.print(F(":"));
        if (gps.time.minute() < 10) {
            myFile.print(F("0"));
        }
        myFile.print(gps.time.minute());
        myFile.print(F(":"));
        if (gps.time.second() < 10) {
            myFile.print(F("0"));
        }
        myFile.print(gps.time.second());
        myFile.print(F("."));
        if (gps.time.centisecond() < 10) {
            myFile.print(F("0"));
        }
        myFile.print(gps.time.centisecond());
    } else {
        myFile.print(F("INVALID"));
    }

    myFile.println();
    myFile.close(); // close file
}

```