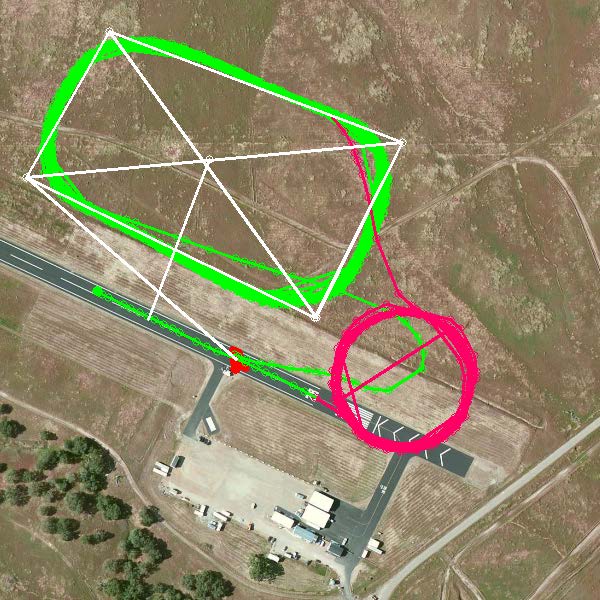
*Lab 5: UAV Flight Data*

**

*This week’s lab will focus on File I/O, handling data, and data structure manipulation. These skills are essential to accomplishing anything with data of any reasonable quantity.*

This week’s lab is an analysis of actual UAV flight data from the NPS Systems Engineering Department. This data was collected from transmissions from 5 UAVs flying simultaneously in an orbit over Camp Roberts airfield. Your task is to perform some basic analytic tasks on the dataset.

**Data:** During these short (~15 minute) flights, each UAV sent roughly 40 Megabytes of data to the ground station, recording basic flight data, system health, etc.. That data has been partially cleaned, combined, and parsed into two files:

* uav\_hud\_data.csv
* ahrs2\_data.csv

You have also been provided with python code file titled “oa2801\_lab5.py”. This file contains two “helper” functions. The first helper function named load\_data() reads the two data sources into a dictionary data structure. The second helper function named time\_convert() can be used to convert a time in HH:MM:SS.SS format (as observed in the two .csv files) and convert it to “minutes after 12:00”. You will need to use this function to convert all of the times in the lab from the HH:MM:SS.SS format (a string) into a minutes after 12:00 format. In this format, the time ‘12:00:00.00’ (a string) returns the value 0.0 (a float) and the time ‘12:30:00.00’ (a string) returns a 30.0 (a float). Note that the load\_data() helper function uses the time\_convert() function to make this conversion for you (provided you choose to solve the lab problem using a dictionary data structure).

**Task 0: Data/Function/Lab Review**

Review the provided data to make sure you understand the data contained in the files. Review the provided helper functions and make sure that you understand how they work. Read the *entire* lab below to ensure you understand the entirety of the analysis you will be asked to perform. **Make sure you understand all the tasks you will need to perform before you begin Task 1 below**; you need to make an informed decision about the data structure(s) you choose during Task 1.

**Task 1: Data Import/Formatting**

Before you can conduct the analysis using Python, you will need to bring the data into Python and store it in an appropriate data structure. This analysis can be accomplished using a List or Dictionary data structure (or any combination thereof). Use the provided load\_data() helper function in the file oa2801\_lab5.py.

**Task 2: Fastest Airspeed**

**Question #1: What is the fastest airspeed achieved for each vehicle?**

A) Write a simple program to calculate the fastest airspeed achieved by vehicle #3 (hint: consider using the sorted() function introduced in the dictionary lecture – this function works on lists and dictionaries). Your code should output something like:

“The fastest airspeed for vehicle 3 was: x.xxx m/s.”

B) Modify your code so that it prints Question #1 above, and then provides the answer for all five vehicles (in order).

**Task 3: Highest Altitude**

**Question #2: What is the highest altitude achieved for each vehicle, and at what time was that altitude achieved?**

A) Write a simple program that calculates the highest altitude achieved by vehicle 1 and the time that altitude was achieved. Your code should output something like:

“The highest altitude for vehicle 1 was xxx.xxxx at time hh:mm:ss.ss”

B) Modify your code so that it prints Question #2 above, and then provides the answer for all five vehicles (in order).

**Task 4: Launch Time**

**Question #3: What is each vehicle’s launch time?**

A) Given the limited flight time of the vehicles, one of the goals is to launch all five vehicles in rapid succession. We can use a groundspeed reading above 10 m/s as a good indication that the vehicle has been launched. **Write a function** named launch\_time() that takes two inputs (the dictionary, a vehicle name) and returns the first time that the groundspeed is reported above 10.0 m/s.

B) Iterate through each of the vehicles and **use your launch\_time function** to calculate the launch time for each vehicle. **Record the results into a list of tuples** with the format of each tuple as (v\_name, launch time). Modify your code so that it prints Question #3 above and then **provides the list of tuples as the answer** (in launch sequence order).

**Task 5: Average Launch Interval**

**Question #4: What is the average launch interval achieved during this experimental run?**

Write the code needed to calculate the average launch interval achieved (the average of the four intervals). Modify your code to print Question #4 above and then provide the answer below.

**Task 6: Close Pass Calculation**

**Question #5: Which vehicle passed closest to (35.7158, -120.7640) and at what time did that occur?**

Sometime during the event, a bystander reported a close pass on a location slightly southwest of the normal operating area. The close pass occurred near the location (35.7158, -120.7640). Write code to calculate the vehicle and time the close pass incident occurred for further investigation. Note that on a small-scale area like this, it is OK to use Euclidean (straight-line) distance measurement vs. “great circle” distance. Modify your code to print Question #5 above and provide the answer.

**Task 7: Batch Processing**

Place all of your code into the script oa2801\_lab5.py. Modify your code as necessary so that when your program is executed, it prints each question (Questions 1 through 5 above) and then the answers to that question. Have your program print all results to the screen.

Generate an output file by running your script from the command line as follows:

**python oa2801\_lab5.py uav\_hud\_data.csv ahrs2\_data.csv > yourlastname\_lab5\_results.txt**

Note that this “redirects” your print statements from the screen and into a new file named yourlastname\_lab5\_results.txt.

**Submission:**

Upload the following files into the Lab 5 Assignment tab in Sakai.

* **oa2801\_lab\_5.py**
* **yourlastname\_lab5\_results.txt**

Note that we will also run your python script from the command line to ensure that it properly outputs a text file that provides the five questions and answers above.