

Due date: 10/28/2014

**Motivation:**

**Emerging trend 1: Project Octocopter:** "Amazon's secret R&D project aimed at delivering packages to your doorstep by "octocopter" mini-drones with a mere 30-minute delivery time", <http://www.cbsnews.com/news/amazon-unveils-futuristic-plan-delivery-by-drone/>, December 2013.

**Emerging trend 2: Express delivery:** Australian textbook rental service [Zookal](#) make good on UAV's (unmanned aerial vehicles) promise to provide lightning-speed personal deliveries. French postal service La Poste claimed to be launching a newspaper delivery drone service in April -- [only to reveal it as a hoax](#). But Zookal looks to be the real deal, offering Sydney students the chance to have their textbooks dropped at any location.

**Emerging trend 3: Project Wing:** Google has built and tested autonomous aerial vehicles, which it believes could be used for goods deliveries. The project is being developed at Google X, the company's clandestine tech research arm, which is also responsible for its self-driving car. Project Wing has been running for two years, but was a secret until now. Google said that its long-term goal was to develop drones that could be used for disaster relief by delivering aid to isolated areas.

They could be used after earthquakes, floods, or extreme weather events, the company suggested, to take small items such as medicines or batteries to people in areas that conventional vehicles cannot reach. <http://www.bbc.com/news/technology-28964260>, August 2014.

Given this emerging scenario consider a concern in today's overcrowded skies with only the large commercial aircrafts. Federal guidelines require craft to be separated by at least 1,000 vertical feet and 3.5 lateral miles.

Incident 1: "The near-miss involving three planes operated by US Airways happened on Tuesday in skies close to Reagan Washington National Airport....The FAA added that due to bad weather, air traffic controllers switched landing and departing operations and miscommunication "led to a loss of the required separation" between the jets. ...The near-miss comes a year after two passenger planes were forced to land without help from the tower at the same airport, after the lone air traffic controller on duty fell asleep....That incident led to a review of staffing levels and worker fatigue at US airports.." August 2012.

Incident 2: "According to a statement issued by Hong Kong's Civil Aviation Department, a Cathay Pacific Airways flight that was bound for Hong Kong International Airport came within 1 nautical mile (2 km) of a Dragonair A330 airplane that was in a holding pattern for landing...The two jets, carrying more than 600 passengers and crew members, came within seconds of colliding, the newspaper reported, citing a former Hong Kong aviation official...." Sept 27, 2011.

Incident #3: "Airliners too close for comfort over New York", CNN, June 25, video, 2013. <http://www.cnn.com/2013/06/21/travel/nyc-jfk-airliners-too-close>

**Problem Statement:** All aspiring "droners" (drone companies) are in discussion with Federal Aviation Authority (FAA) to set standards and policies to allow them to achieve their drone dream. But just imagine the issues and challenges in allowing drones in the sky that is already crowded. **Let us assume that** Federal Aviation Administration (FAA) wants to address these and such other problems in due to drones, planes, buildings and people, navigation of terrain, obstacle detection and avoidance, esp.in the context of many high tech companies entering the space with their own drone fleet. Understand most of today's aircraft's have explicit collision detection and alarm systems and there are strict rules about how close they can come to another flight.

One of the approaches we (CSE321ers) want to recommend is a combination of sensors on these drones, collision detection system and a **real-time system** that automatically controls the drone landing, take-off and cruising. **Your task is to design, implement and test this realtime software** we will call "**Drone Control and Deployment for social good (DECOY)**"; assume that the sensors and collision detection systems are in place and we need to develop the software. Alternatively collision detection can be determined by some mutually agreed upon GPS/position based **algorithm**.

**Study of existing systems:** Of course, for the larger crafts than the drones, there is a system called Traffic Collision Avoidance System (TCAS) that is based on sensors and instrumentation placed on each airplane in service. However we feel it is local solution centered on the aircraft and does not provide a global view of the whole picture for the drones since drones are operate on completely different "plane" and are remotely controlled; [http://en.wikipedia.org/wiki/Traffic\\_collision\\_avoidance\\_system](http://en.wikipedia.org/wiki/Traffic_collision_avoidance_system)

#### **Requirements:**

1. DECOY is installed and operates at the drone control center (distributed all over the drone space) and monitors and controls the operation of the drones landing and taking of at these centers and if needed at their destinations.
2. Once the drone leaves the area /vicinity of a control center and reaches a cruising altitude the monitoring is handed-over to an intermediate DECOY station. So DECOY system is indeed a truly distributed realtime monitoring system. (This handover from one DECOY center to another is similar how you cell phone is handed over from one cell tower to another.)
3. You need to worry about only one DECOY center. It is possible that the model of operation for the DECOY is that it totally controls its assigned fleet from takeoff, cruise, and navigate obstacles and deliver and return.
4. Regular flights have runways for takeoff and landing. We are assuming vertical takeoff and landing for the drones. However only one drone can take off from a center/station at any time. Same with landing. That means a drone is assigned the task, it picks up the payload, and then request clearance to take off. Once it is given it takes off.

5. DECOY software also keeps track of number of drones, whether they are active and their state. Assume that a runway is a mutually exclusive resource that can allow a take-off or a landing at any given time.
6. When a drone requests landing, if a landing space is available it is allocated. If not, the drone keeps hovering around in a holding pattern. All the drones requesting landing are queued up and are served first come first serve basis.
7. Collision detection instrumentation (sensors, alarms etc.) will detect imminent mid-collision, runway collision, and any collision within the DECOY station when the drones are moving around. Appropriate action to be taken when collision detection happens.
8. Life cycle of a drone: (These are the states) DECOY registers an active drone, it is at gate, gets clearance to take-off gets clearance for takeoff, takes-off, reaches cruising altitude, delivers payload and returns. Exceptional situations are collision detection and the corrective operations that may happen at various points.
9. Assume other conditions and make sure you document your assumptions.
10. Design this realtime system, implement it and test it with a system handling, say, f (10) drones and single station. Then test it with multiple stations (long distance drones).

**Design:**

1. Start by designing the state diagram representing the life cycle of a drone. Also use a table-driven scheduling for the flights monitored by a DECOY. You will have to design these two before you can start implementing the code. You may have to revisit and revise your state diagram and the state transition table based on the needs.
2. Implement the DECOY system using C/C++ language. Define a function for each of the operations. Assume collision detection as an asynchronous event. (How will you simulate an interrupt and handle it ?)
3. You can use multi-processing for simulation of several drones co-existing.
4. Test your system for various conditions using sample scripts.
5. For implementation of the operation you may print out the message and allow some predetermined elapsed time for the operation taking place.
6. On completion your program should run DECOY application; should take sample scenarios as input and play out the scenarios and action taking place through a series of messages printed out.
7. You need a mechanism for realizing /implementing parallelism, mutual exclusion (collision avoidance and shared data), communication between the drone and the controlling station, and others.

**What to submit?** Please submit your design documents, well commented code online and clear simulation run outputs showing various conditions: submit cse-321 your files

You will have to demo your project to the TAs before you get your grade.