COEN 320 Project

This project involves all parts of the design, implementation, testing, and analysis of a simplified real-time system for air traffic monitoring and control (ATC) system. It must be written in C or C++, and implemented, tested, and analyzed in a normal PC running the QNX real-time operating system.

1. Air Traffic Monitoring and Control (ATC)

ATC is a pertinent example of a complex real-time system, which contains different sub-systems that must be robust, fault-tolerant, safe, secure, scalable, and required to work in a real-time manner. An ATC is aimed at the control of the airspace at any time and circumstance to enable the safe movement and navigation of aircraft in the airspace.

The goals of an air traffic control system are i) to maintain reasonable and orderly air traffic flow and separation between aircraft in each air domain, aimed at safe movement of aircraft in any circumstances, and ii) to prevent collision between aircraft and aircraft from crashing into obstacles on the operating ground. The ATC system involves the hard work and collaboration of many specialists. The tower, approach, and en-route specialists are three categories of interest in this project. Besides, ATC contains an elaborated set of equipment and sub-systems for highly automated communication, detection, and visualization.

In the following paragraphs, an ATC system is presented in more detail. The presented discussion is from the thesis "A teamwork-oriented air traffic control simulator" by Mounir Sidhom, which can be downloaded at: https://core.ac.uk/download/pdf/36696446.pdf

The basic control points in an air traffic control system include three main independent cooperative activities shown in Fig. 1:

- Tower control area;
- Approach or Terminal Radar Control (TRACON) area;
- En-route Area.

The airspace is divided into volumes, called sectors. Each one of the three entities mentioned above is responsible for traffic control in a sector. The control of an aircraft passes from one controller to another as the aircraft moves from one sector to another.
The tower control is responsible for the control of any movement of ground vehicles and flying objects within the area of the airport itself and an airspace volume with a radius of 2 to 30 miles and a height of 1,500 to 2,000 feet centered in the airport. Airport Tower Controllers regulate a specific airport's traffic. They are responsible to give pilots permission to take off and land. They also direct ground traffic, which includes taxiing aircraft, vehicles, and airport workers.

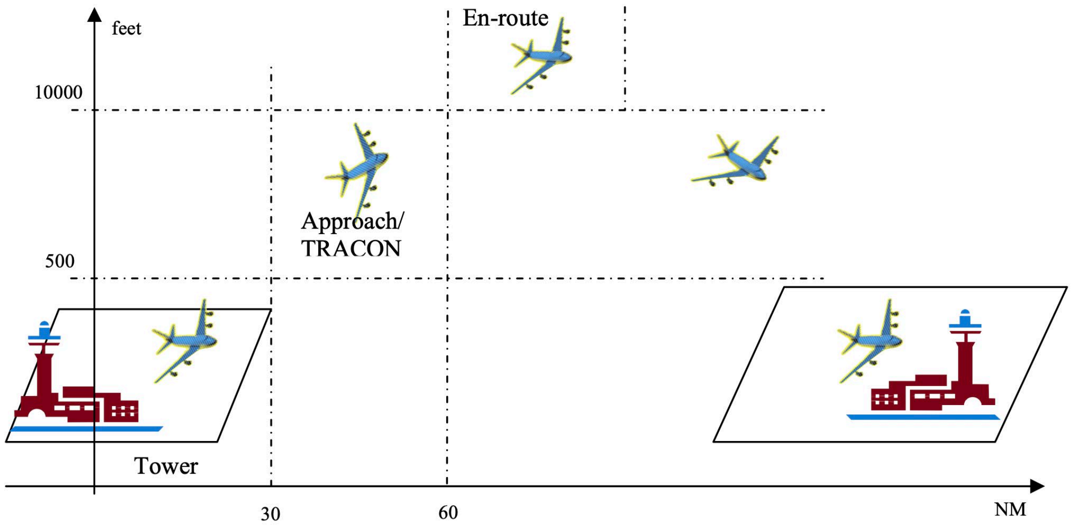

Fig. 1: Air traffic control system

The TRACON is in the vicinity of a large airport and controls aircraft within an airspace volume within a radius of 30-50 nautical miles (56 to 93 km) from the center of the airport, and at a height between the surface and 10,000 feet above the airport. Generally, the TRACON controls departures, arrivals, overflights, and aircraft operating under Visual Flight Rules (VFR) or Flight Instrument Rules (IFR) traffic flows. As discussed in Sidhom, 2006:

> *"Departure aircraft are handed off from the tower to the TRACON when they are between 1,000 feet to 2,000 feet high, climbing to a pre-determined altitude. The TRACON controller working this traffic is responsible for clearing all other TRACON traffic and, based on the route of flight, placing the departing aircraft on a track and in a geographical location (sometimes referred to as a "gate") that is pre-determined through agreements for the en-route center controller. Arrival aircraft are handed off from the en-route center in compliance with pre-determined agreements on routing, altitude, speed, spacing, etc. to the TRACON center. The TRACON controller working this traffic will take control of the aircraft and handle it with other aircraft entering the TRACON from other areas or "gates" into a single file or final for the runway. This spacing is critical to ensure the aircraft can land and clear the runway prior to the next aircraft touching down on the runway. The tower may also request expanded spacing between aircraft to allow aircraft to depart or to cross the runway in use. Over-flight aircraft are aircraft that enter the TRACON airspace at one point and exit the airspace at another without landing at an airport. They must be controlled in a manner that ensures they remain separated from the climbing and descending traffic that is moving in and out of the airport. Their route may be altered to ensure this is possible. When they are returned to the en-route center, they must be on the original routing unless a change has been coordinated."*

The En-route control controls aircraft when they leave the TRACON volume and reaches their cruising speed and altitude. En-route control set of separation standards that define the minimum distance allowed between aircraft; these distances vary depending on the equipment and procedures used in providing ATC services.

2. Simplified version of to be implemented of the ATC

The goal of this course project is to implement and analyze a simplified version of an ATC system. The ATC simplified version to be implemented shall control aircraft flows in the en-route control area. The en-route control handles aircraft movement from the moment they are “handed-off” to the en-route ATC, by its controlling TRACON neighbor site, to the moment they leave the en-route control airspace, i.e., when they are “handed-off” by the en-route ATC to a neighboring TRACON ATC.

The airspace controlled by the considered en-route ATC is a 3D rectangle area that is 15,000ft above the sea. This area is assumed to be 100000 x 100000 in the horizontal plane and 25000 in height, as shown in Fig. 2.

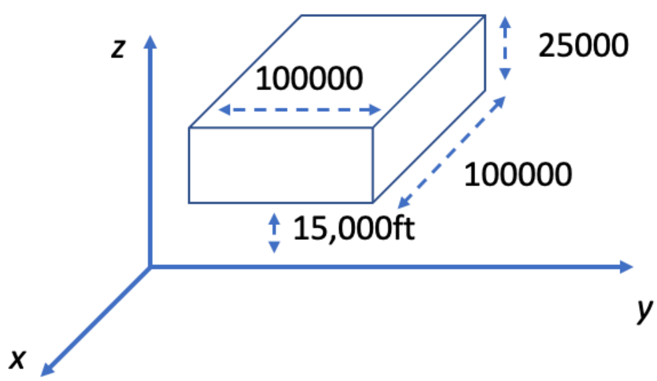

Fig. 2. 3D airspace

An aircraft enters the considered area flying in a horizontal plane at a constant velocity. It maintains its speed and altitude unless directed by the ATC to change. We will not be concerned with the details of the radio communication subsystem that allows the ATC controller and pilot to speak directly.

One of the main goals of the simplified en-route ATC to be implemented is to maintain adequate aircraft separation distances. An aircraft must have a distance no less than 1000 units in height and 3000 units in width/length from another aircraft. If they fall within that minimum separation, the ATC system must notify the controller, which will in turn notify the aircraft to adjust either its speed or position.

3. Project specification and possible software architecture

The implementation of the simplified en-route ATC must perform the following functions:

- Display a plan view of the space every five (5) seconds showing the current position of each aircraft.
- Check all aircraft in the airspace for separation constraint violations at current_time + n seconds, where n is an integer parameter. The controller must be able to change the parameter n at runtime to react to the degree of congestion in the space.
- Emit an alarm if a safety violation is found or if a safety violation will happen within 2 minutes. This can be a sonorous alarm or a visual notification on the screen.
- Store the airspace in a history file every twenty (20) seconds. There should be enough information in this log to generate an approximation of the history of the airspace over time.
- Store the operator requests and commands in a log file.

The en-route ATC subsystems and the communication among them are shown in Fig. 3 and discussed below.

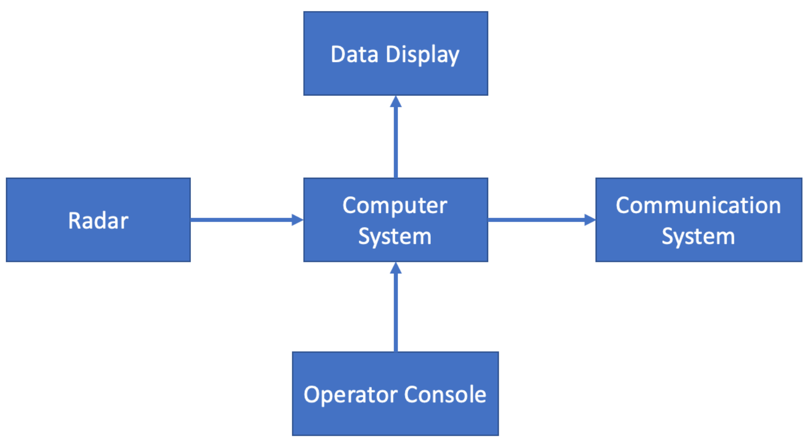

Fig. 3. Components of the simplified ATC to be implemented

Primary surveillance and secondary surveillance radars: The primary surveillance radar (PSR) is a pulsed beam of ultrahigh-frequency radio waves in a circle from a rotating aerial. Its main output is a spot in the screen representing "illuminated" objects, i.e., objects in the airspace that reflected some energy from radar emitted beam, placed according to the distance of the object to the center of the radar. The secondary surveillance radar (SSR) emits special signals, called interrogation signals, to any illuminated object by the PSR. Transponders at the aircraft will respond to interrogation signals. The responses are then processed, and the following details are displayed on the screens of air traffic controllers:

- The flight ID
- The aircraft flight level (FL)
- Aircraft’s speed
- Aircraft’s position

The obtained information of each aircraft must be passed to the computer system.

Computer System: It is responsible for periodic computations to determine if there is any violation of the airspace separation constraint. It is also responsible for the alert/notification of such event to the operator. This component determines if there is (or will be) a safety violation and emit an alert to notify the operator. Moreover, it will send to the Data Display the ID and position of the aircraft to be shown on the screen of the controller.

Operator console: Enables the controller to send commands to the aircraft. The command that can be sent is to request the aircraft to change its speed, altitude, and/or position. Moreover, the operator console can be used by the controller to request augmented information about a specific aircraft to be shown in the radar display. For instance, let’s assume that aircraft #1 is in the airspace. The controller can use the operator console to request that the flight level, speed, and position of aircraft #1 be displayed on the screen. To do so, such a request must be sent to the Computer System, which will in turn send the additional information of #1 to be shown by the Data Display.

Communication system: This subsystem is responsible for the transmission of controller commands to the specified aircraft. To send a command m to an aircraft R over the communication subsystem, the computer system emits the following command: send(R,m).

4. Implementation requirements

The following implementation requirements must be met:

- Fill in the missing details of the environment, inputs and outputs, and software functions and data. An input file shall contain the information of the aircraft and shall be used in the process of aircraft entering the en-route monitored area. Each entry in the input file shall be an aircraft with the following details:
    - Time, ID, X, Y, Z, SpeedX, SpeedY, SpeedZ
        - Time: It is the moment in time that this aircraft should appear within the boundaries of the area.
        - ID: Aircraft ID.
        - X, Y, Z: Coordinates the entering aircraft at the boundaries of the area.
        - SpeedX, SpeedY, SpeedZ: Speed of the aircraft in each coordinate dimension.
- Each aircraft shall be implemented as a periodic task (process or thread). It shall update its location every second, from its speed and previous location. It shall implement a function that will answer the radar requests by sending its ID, speed, and position every time the radar requests them.

- The radar operation must be simulated as follows. It must learn the presence/location of each aircraft in the monitored space by communicating directly with the thread/process of each aircraft.
- The active part of the system should consist of a set of periodic tasks, where periodic polling is used to handle sporadic events. All processes or threads share a single processor.

- **Inter-Process Communication and System Architecture Requirements**

The system must be implemented as separate QNX executables for each major subsystem, utilizing **shared memory** for inter-process communication (IPC):

Radar Subsystem: Integrates primary and secondary surveillance radar functionality.
Computer System: Performs safety checks, generates alerts, and manages command routing.
Operator Console: Handles the user interface and controller inputs.
Data Display: Visualizes airspace and aircraft information.
Communication System: Simulates the transmission of commands to aircraft.
All communication between the above subsystems must be conducted using POSIX shared memory (shm_open(), mmap()) for data exchange. Each executable must operate as an independent QNX process with appropriate memory synchronization mechanisms to ensure data consistency and integrity.

- Test your system under various operating conditions (low, medium, high, and overloads). The system load is determined primarily by the number of aircraft in the space, the degree of congestion in the space, and the amount of IO traffic.
- Measure the execution times of each process (best and worst case if possible). Test for scheduling feasibility using rate monotonic fixed-priority assignment.
- Must use the QNX Software Development Platform 7.0 (or later version) for x86 Targets.
- Using any IDE other than QNX Momentics will not be accepted.
- Running the system in any OS other than QNX RTOS will not be accepted.

5. Final team deliverables

Each team must submit by the project deadline the following:

- A copy of the well-commented source and a set of samples, i.e., input files, for testing on the QNX real-time platform.

No less than 10 pages of report specifying a high-level description and modeling of the implemented ATC system. The project report must include a detailed diagram showing the shared memory architecture and data flow between processes. The report should also include a good summary of the results and the **contributions made by each member of the team**. The report guidelines will be posted on Moodle.