Control Tower System Analysis

Table of Contents

Introduction and Working Hypotheses 1
- System Description 1
- System Behaviour 1
- Working Hypotheses 1

System Modelling 2
- Queuing Theory Model (attempt) 2
- Simulation Model 3

Preliminary Analysis 4
- Stability Condition 4
- Warm-up Time 5

System Parameters Effects 6
- Utilization Factor ρ 6
- Parking Time t_P 6

System Insights 7

Introduction and Working Hypotheses

System Description

The system under analysis consists in an airport with a single runway, which can be used by one airplane at a time for landing or take-off operations, a parking area, where airplanes are temporarily stationed between landing and take-off, and a control tower, which routes the air traffic within the airport.

System Behaviour

The system behaviour can be described as follows:

1. Airplanes intending to land reach the airport with an interarrival time "t_A".
2. Whenever an airplane intending to land reaches the airport, it enqueues for landing waiting for the authorization from the control tower.
3. As soon as authorized by the control tower, the airplane performs the landing operation occupying the runway, which completes in a time "t_L".
4. As soon as the airplane has finished landing it frees the runway and moves towards the parking area, where it will remain stationed for a time "t_P".
5. When the airplane finishes its parking time, it enqueues for take-off, again waiting for the authorization from the control tower.
6. As soon as authorized by the control tower, the airplane performs the take-off operation, occupying the runway, which completes in a time "t_O".
7. When the airplane completes the take-off operation, it leaves the system.

From here the control tower routes the traffic within the airport by authorizing the landing or take-off of the airplane having waited the longest to use the runway, assigning it to the next longest waiter as soon as the airplane completes its landing or take-off.

Working Hypotheses

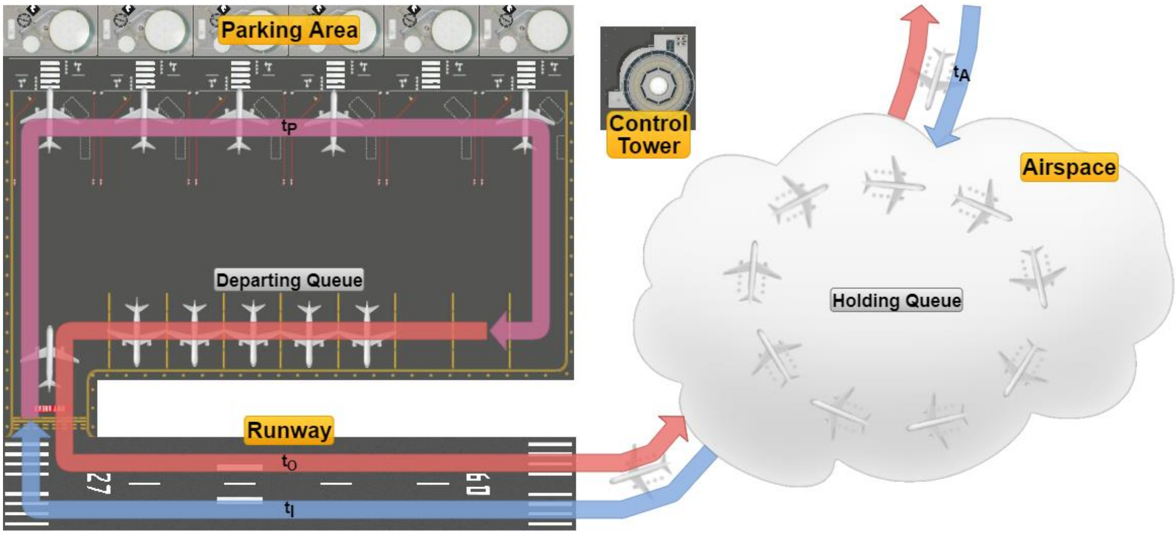
The system must be analysed under the following given hypothesis:

1. The system must be analysed supposing the "t_A" , "t_L", "t_P" and "t_O" times described above both as constants (deterministic regime) and as rates of exponential random variables (stochastic regime).
2. The airplanes awaiting landing have an infinite fuel supply, meaning that they can wait for an infinite time without the risk of crashing.
3. The parking area has an infinite airplane capacity.

The system analysis will also be based on the following additional hypotheses:

4. The system will be analysed starting from an empty state, meaning that there are no airplanes parked, landing or taking off, and where the first airplane will reach the system in a time "t_A".
5. The airplanes parking time "t_P" starts as soon as they leave the runway, and comprises the time required to reach the parking facilities, to perform any passengers/cargo unloading/loading and refuelling, and to leave the parking facilities reaching a separate area adjacent to the runway, where they will wait for the authorization to take-off from the control tower. Following this description, the total number of grounded planes within the airport is given by the number of parked airplanes plus the number of airplanes enqueued for take-off.
6. The system time evolution strictly attunes to the behaviour described above, where real case delays such as the ones determined by the communications between the airplanes and the control tower, the ignition time of the engines prior to take-off, or the local spatial displacements of the airplanes awaiting landing or take-off are not taken into account.

System Modelling

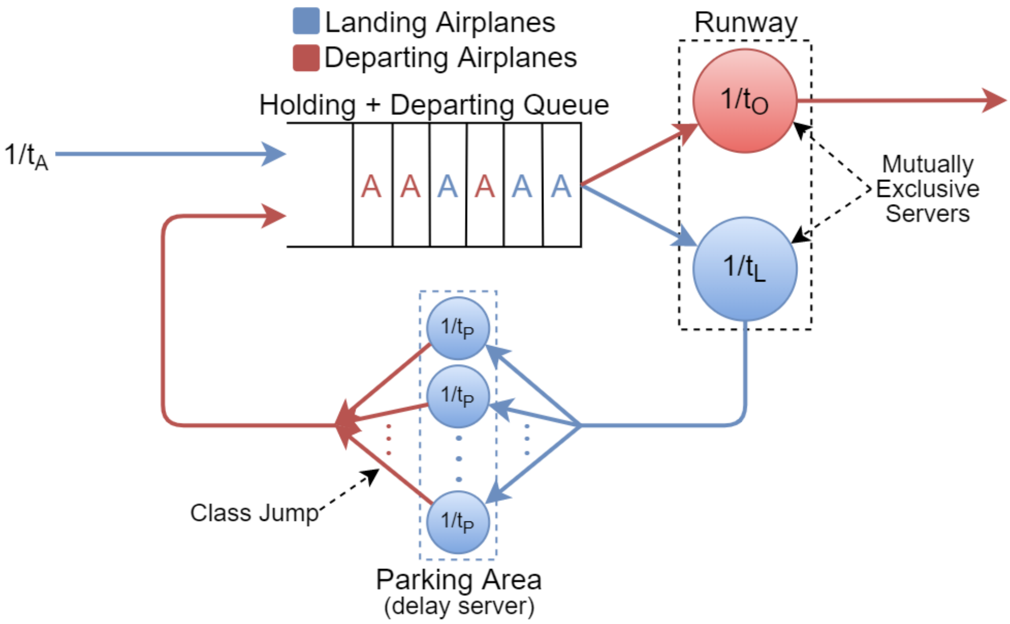


The system can be functionally divided into the following components:

- The *Airspace* surrounding the airport, where the airplanes intending to land arrive with an interarrival time "t_A" and enqueue for landing in the *Holding Queue*, and where airplanes transit once they have taken off, leaving the system.
- The *Runway*, which is used mutually exclusively by airplanes for landing and take-off operations, which are performed in times "t_L" and "t_O" respectively.
- The *Parking Area*, which consists of the facilities where the airplanes transit through after they have landed and before they are ready for take-off, which occurs in a time "t_P", after which the airplanes enqueue in a separate *Departing Queue* adjacent to the runway waiting for the authorization to take-off.
- The *Control Tower*, which acts as a logical entity routing the traffic within the airport.

Queuing Theory Model (attempt)

The system can be tentatively described in terms of queuing theory as a classed routing network with the jobs representing airplanes divided into the two classes as follows:



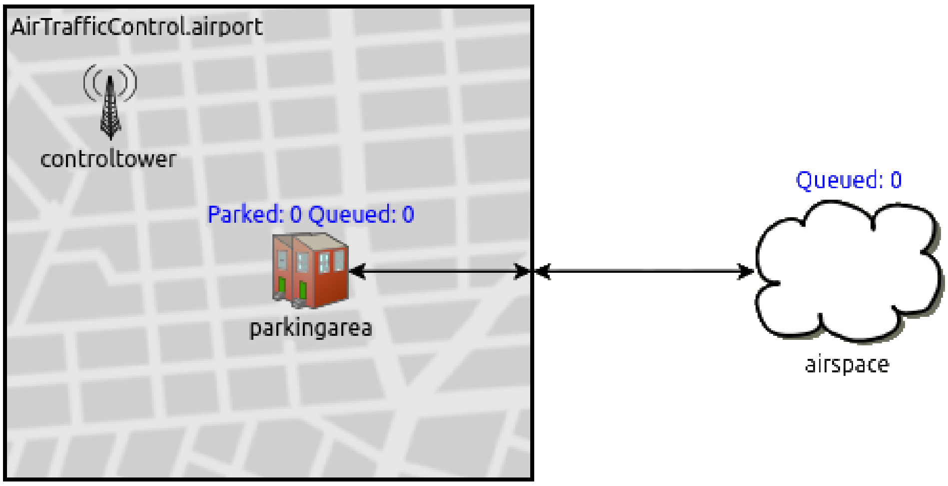
Where:

- The *Holding* and *Departing Queues* have been logically merged into a single virtual queue having as server the *Runway*, which in turn is divided into two logical servers with service rates “$\frac{1}{t_L}$“ and “$\frac{1}{t_O}$“ relative respectively to the landing and departing classes, servers whose services are mutually exclusive (i.e. just one airplane at a time can be served in the runway by the virtual server associated with its class).
- The *Parking Area* represents a Delay Server with service rate “$\frac{1}{t_P}$”, which also switches the airplanes from the Landing to the Departing class.
  It should also be noted that, representing a Delay Server, the *Parking Area*’s parking time “t_P” will have no effect on the system’s stability, as is thoroughly discussed later in the document.
- The *Control Tower*, being a logical entity, finds no correspondence in the model.

From here, since we are unable to determine the steady-state equations of the network and thus its performance metrics, in order to analyse the system the use of a simulation software is required.

Simulation Model

The simulator software used is OMNeT++ 5.5.1, wherein the system model was reproduced as follows:



Where:

- The *airspace*, the *parkingarea* and the *controltower* represent simple modules, the last two being logically grouped inside an *airport* compound module representing the airport grounds.
- The *runway* represents the connection between the *parkingarea* (and the *airport* module) and the *airspace*.
- The *controltower* acts again as a logical module that doesn’t exchange messages (i.e. airplanes) with the others, and being the communication delays between the tower and the airplanes not taken into account (hypothesis 6.), the synchronizations between the landing/departing airplanes and the control tower are performed via cross-module calls, where each time a landing/take-off is completed the control tower assigns the runway to the airplane with the longest waiting time in the *Holding Queue* (*airspace*) and the *Departing Queue* (*parkingarea*).

The results that follow were obtained through the sampling of the following quantities during the system time evolution:

- The system total response time (*AirportResponseTime*).
- The airplanes’ waiting times in both queues (*HoldingQueueWaitingTime* and *DepartQueueWaitingTime*).
- The number of airplanes waiting in both queues (*HoldingQueueSize*, *DepartQueueSize*).
- The number of parked airplanes (*ParkedPlanes*).

Preliminary Analysis

Stability Condition

Following our tests, we determined the stability condition of the system to be:

$$\boxed{t_A > t_L + t_O}$$

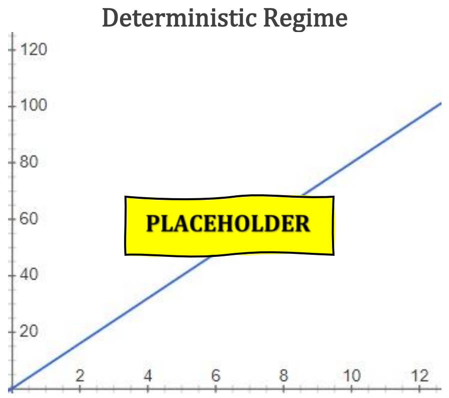
Which expressed in terms of interarrival rate $\lambda_A = \frac{1}{t_A}$ and service rates $\mu_L = \frac{1}{t_L}$ and $\mu_O = \frac{1}{t_O}$ becomes:

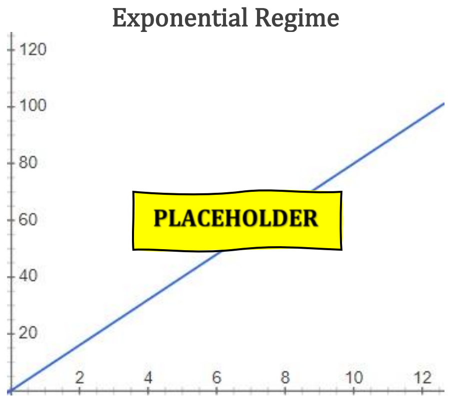
$$\boxed{\lambda_A < \frac{\mu_L \mu_O}{\mu_L + \mu_O}}$$

Meaning that the overall system presents an equivalent service rate of:

$$\mu_E = \frac{\mu_L \mu_O}{\mu_L + \mu_O} < \mu_L, \mu_O \quad \mu_L, \mu_O > 0$$ [1]

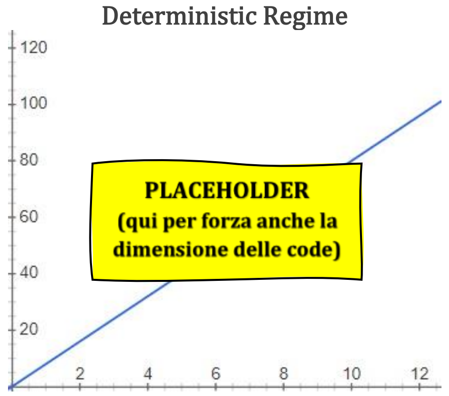
System Instability ($t_A < t_L + t_O$)

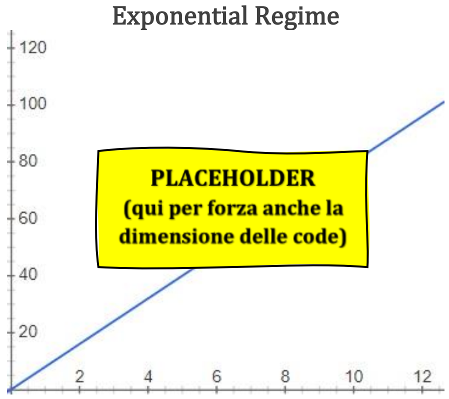




As shown by the plots above, if $t_A < t_L + t_O$ the *AirportResponseTime* [e anche le dimensioni delle due code, se volete plottarle] diverges in both regimes, compelling evidence that the system is unstable.

System Stability ($t_A > t_L + t_O$)





If $t_A > t_L + t_O$ we can observe that:

- In Deterministic Regime the mean size of both queues drops to "0" while their maximum size caps at "1", which as better discussed later is due to the conflicting utilization of a single

[1] A mathematical analogy of the expression of the equivalent service rate can be found in electric networks theory as the equivalent conductance of the parallel of two conductances $G_{EQ} = G_1 // G_2 = \frac{G_1 G_2}{G_1 + G_2}$

shared resource (the *Runway*) by two modules (The *Holding* and the *Departing Queues*). [Verificare, aggiungere roba sull'AirportResponseTime].

- In Exponential Regime we can preliminarily observe the quantities to be strongly correlated, both with themselves (*autocorrelation*) and with the others, in particular the *Waiting Time* and the *Size* of both queues and the *Waiting Time* of both queues with the *AirportResponseTime*. [Verificare, aggiungere, correggere].

From here, since the statistics do not diverge as the sample size and so the simulation time increases, we can assert the system to be stable, thus proving its stability condition.

Warm-up Time

<u>NOTA</u>: Questa sezione l'ho scritta molto ad "intuito", e sebbene abbia una buona confidenza su quanto scritto, deve essere ovviamente tutto verificato.

The system warm-up time was calibrated towards the moment in which the system starts exhibiting its most characteristic feature, that is the conflicting utilization of the *Runway* by the two queues, which first happens after the first airplane concludes its parking time and so enqueues in the *Departing* Queue, which occurs at the mean time of $t_A + t_L + t_P$.
At this moment we can observe that:

- In Deterministic Regime under the stability condition, since no airplane has yet taken off, the "transient" contribution of the departing service time "t_O" to the overall service time will be null, from which by taking back the stability condition:

$$t_A > t_L + t_O \;\Rightarrow\; t_A > t_L \;\Rightarrow\; \lambda_A < \mu_L$$

  Therefore at time $t_A + t_L + t_P$ there will be no planes enqueued in the *Holding Queue*, one airplane might be landing, and apart from the limit case of a new airplane entering the *Holding Queue* at the exact same time, the airplane exiting from the *Parking Area* will be the next to use the runway.

- In Exponential Regime due to the randomness of the parameters the previous assertions are generally invalid, where the first airplane concluding its parking time at the mean time $t_A + t_L + t_P$ can come upon a non-empty *Holding Queue*, and so won't be the first to use the runway, situation that causes an initial disparity in the *Waiting Time* and the *Size* of the two queues. [which probably fades in time, verificare con grafici]

From here, to account for the variability the $t_A + t_L + t_P$ time in exponential regime we further added a multiplicative constant, selecting a warm-up time of:

$$\boxed{\text{Warm-up Time} = 5(t_A + t_L + t_P)}$$

System Parameters Effects

[In questa sezione farei più sottosezioni in cui in ognuna in condizioni di stabilità si fissano tutti i parametri (t_A, t_L, t_P, t_o) tranne uno, e si studia come il variare di quello influisce con gli altri utilizzando diversi seed, determinando intervalli di confidenza e tutto.
Secondo me data la condizione di stabilità il variare di t_A, t_L e t_o (o λ_A, μ_L e μ_O) avrà gli stessi effetti, che studierei congiuntamente facendo variare l'utilization factor "$\rho = \frac{\lambda_A}{\mu_E}$" del sistema, mentre il "t_P" è un parametro che sicuramente va studiato a parte (ed essendo un delay server secondo me influirà solo sul warm-up time e al massimo per la storia del conflitto sull'accesso alle code)]

Utilization Factor $\rho = \frac{\lambda_A}{\mu_E}$

[...]

Parking Time t_P

[...]

System Insights

[In questa sezione invece studierei i comportamenti più avanzati del sistema, magari con parametri fissi e ovviamente molti seed, tra cui:

- Qualche parola sul conflitto delle due code nell'accesso alla risorsa condivisa e come questa influisce sulla loro Size e Waiting Time
- La correlazione tra i parametri (auto e non) e come viene affrontata (subsampling?)
- Il fatto che l'andamento della Size della DepartingQueue è la stesso di quella della HoldingQueue sfasata nel tempo (da confermare, si vede con sample piccoli, mentre il Waiting Queue non sembra influenzato).
- Il fatto che il sistema tende a tornare periodicamente allo stato iniziale.
- [Altro?]