Control Tower (1)

Meloni Enrico
Ruisi Davide
Zippo Raffaele

Index

1. System and objectives analysis

1.1 System description

The target system to study is an airport with a single runway, used both for landings and takeoffs. At any given time, only one airplane can occupy the runway. Each airplane follows these steps to enter and leave the airport:

1. It queues for landing;
2. It lands on the runway, taking time t_l;
3. It moves in the parking lot and stays there for a time t_p;
4. It queues for take-off;
5. It takes-off, taking time t_t, and then leaves the airport.

The queues are FIFO, and their capacity is assumed to be unlimited. The parking lot too is assumed to have unlimited capacity. Airplanes are considered to have unlimited fuel, so they can wait flying for an unlimited time before landing.

A control tower manages the traffic through the airport, applying the following policy when the runway is unoccupied:

- Serve an airplane queued for landing.
- If the landing queue is empty, serve an airplane queued for take-off.

1.2 Objectives and performance indices

Objective of our study is to gather useful insights about the airport system. Regarding the queues, we decided to study the distribution and fairness of the waiting times experienced by airplanes. In the system, we suppose the capacities to be unlimited, which of course cannot be true for a physical airport. So, we decided to study the used parking lot space, in order to determine a sufficient size given the flow of airplanes through the parking lot. It's similarly useful to determine the required space for airplanes waiting to take off. In this system airplanes have infinite fuel; in a physical system airplanes have instead a finite quantity of fuel, which is the amount needed for flying to the destination plus an excess due to the in-air waiting time before landing. The less fuel an airplane carries, the more efficient it will be. So we decided to study the waiting time in the landing queue, quantity that can be used to estimate the excess fuel needed.

2. Modeling

2.1 Indices and factors evaluation

Based on the objectives set for the study, we observed the characteristics of the system that the model needs to show, with respect to the indices that we want to measure. We identified the required indices as the following:

- Waiting times of both queues
- Length of take-off queue
- Number of airplanes in the parking lot

To evaluate the state of the system, that is distinguishing the transition state from the steady state, we also added as an index the throughput of the runway both for airplanes landing and taking off. We then considered the factors for configuration of the system, that are:

- The times t_l, t_t and t_p
- The arrival rate of airplanes, λ.

2.2 Model structure design

From the observed requirements, we can restrict the characteristics to model in two categories:

- The path taken by airplanes through the airport
- The control system that applies the policy

The path taken by the airplanes consists of the landing queue, the runway, the parking lot, the takeoff queue and then the runway again. These steps can be modeled as nodes through which airplanes move.

The control system consists of a control tower that, knowing the state of the airplanes in the system, issues permits to continue. We assume that, for our study, the delays introduced by communication between the control tower and the airplanes are negligible. This part can be modeled as a node, the control tower, which receives updates from the path nodes and process them to give to the appropriate queue node the permission to send an airplane forth.

Following on these considerations, we modeled the system as in [Figure 1].

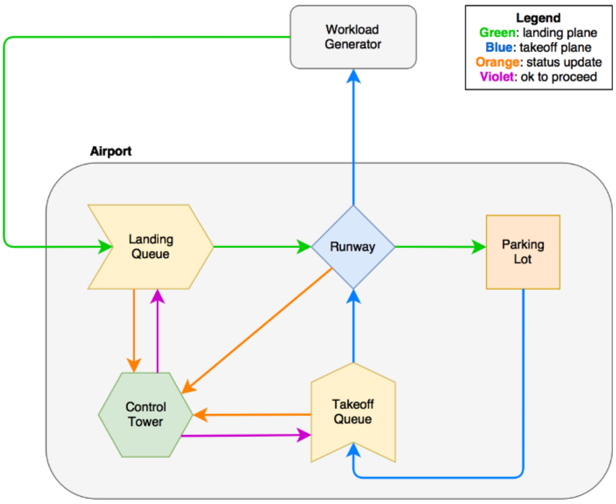

FIGURE 1 - MODEL OF THE AIRPORT SYSTEM

Important simplification in the model with respect to the system is that airplanes are passive actors in the appliance of the precedence policy, while the path stages are the active actors together with the control tower.

Airplanes are modeled as messages, while the stages the airplanes follow through the airport are modeled as nodes exchanging airplane messages. These stage nodes also exchange control messages with the control tower.

Control messages can be of two types: *status update* and *ok to proceed*. Status updates are issued by queues and by the runway to inform the control tower of their current state. Ok to proceed messages are sent by the control tower to the queues. The control tower applies its policy each time it receives an update. The runway can be in one of three states: free, occupied by a landing airplane, occupied by a leaving airplane. Each time it becomes free it sends a status update to the control tower. The queues can hold an infinite number of airplanes, and send a status update to the control tower each time a new airplane arrives. When an *ok to proceed* message is received from the control tower, the first airplane in the queue is sent to the runway.

The parking lot holds each airplane received for a time t_p and then sends it to the take-off queue. The *workload generator* models the external environment, it sends landing airplanes to the airport and it receives leaving airplanes.

3. Implementation

We implemented the model with the OMNeT++ framework. We chose different strategies to store *Plane* messages in stage nodes:

- A *PlaneQueue* stores *Plane* messages in a STL *queue* container;
- *ParkingLot* doesn't store parked *Planes* in a data structure, relying instead on *scheduleAt()* to simulate the waiting time spent by each *Plane*;

To implement the policy of *ControlTower*, we need to distinguish whether the *Runway* is occupied or not when the status update about an enqueued *Plane* is received. If the *Runway* is occupied, a counter is incremented to keep track of the queue status. When the *Runway* becomes free again, the counters are checked to see if there are airplanes in queues and an *OkToProceed* message is issued if that's the case. The counter corresponding to the landing queue is checked and eventually served first, to give priority to the landing airplanes.

This structure has however a weakness related to the software implementation, that may lead to inconsistencies with respect to the model. In the model, events happening at the same time are also observed at the same time. This is of course not applicable to software, where messages received at the same simulation time must be inspected sequentially. This may lead to a situation where status updates received at the same simulation time are processed by the *ControlTower* in such an order that the takeoff airplanes are served first, breaking the policy. To solve this issue, we used the *cMessage::setSchedulingPriority()* method, which allows to set a strict order of processing between messages. The order of message processing we chose across the whole simulation is the following:

0. Plane messages
1. Status updates messages from the landing queue
2. Status updates messages from the takeoff queue
3. Status updates messages from the runway
4. Ok to proceed messages
5. Periodic sensing tasks

With this order of processing the control tower will correctly follow the policy under any condition. Sensing tasks are executed last so they collect all the info about the events happened up to the given simulation time.

4. Warm-up time

We decided to use the system throughput as the metric to evaluate when the steady state is reached. In fact, we know that any system such that no jobs are created within it has a steady state mean throughput equal to the mean arrival rate. We can so analyze the throughput of the simulations to find an upper bound for the time required by the throughput to become stable around the mean arrival rate. An example of such analysis is the graph in [Figure 2].

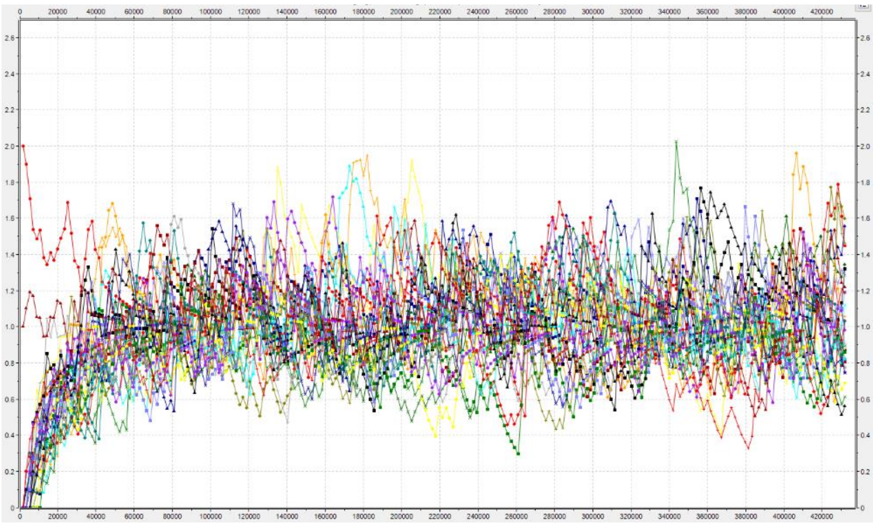

FIGURE 2 - MEASURED THROUGHPUT OF INDEPENDENT REPLICAS OF THE SAME SCENARIO

This graph is obtained by counting the airplanes that leave the system during periods of fixed length, set equal to the mean interarrival time, and then applying an exponential mean to reduce noise and have a stable measurement. The different colors correspond to independent replicas of the scenario.

In the steady state, the mean measured throughput should be 1. We can quickly observe in the example that this happens after a time of 60.000 seconds, that is ~17 hours. This time is the warm-up time to be used when extracting data out of this scenario. In fact, the warm-up time is related to the scenario and this study was done for each new scenario to be analyzed. From there on, we will omit this computation and the warm-up times used for our analyses.

5. Verification and validation

5.1 Verification

To ensure the correctness of the simulation software, we first compared it to the model. We picked deterministic scenarios, and we used the graphical execution environment provided by the OMNeT++ IDE to follow the messages exchanged through the network and compare them to a manual execution of the model. The scenarios we choose are described in next paragraphs, as we used them also to validate the results of the software.

5.2 Validation

Once we verified the software against the model, we went on to validate its results against the system. We used deterministic and exponential distributions for the times as, with the limitations discussed below, both allowed to analytically compute results to be compared to the ones of the software.

5.2.1 Deterministic case

First, we studied the model under deterministic times.

- $\frac{1}{\lambda} = t_l = t_t = t_p = k$

In this case the system is unstable, as the take-off queue grows indefinitely while the landing queue is always empty. This is because the landing time is equal to the interarrival time, so as soon as the runway is left free it's occupied again by a new landing airplane and take-off airplanes never get the chance to leave.

- $\frac{1}{\lambda} = 2k; t_l = t_t = t_p = k$

In this case, there's time for a landing and a take-off between the arrival of two airplanes. The system is stable with a warm-up time of 2k and a period of 2k after that. In the first half of the period, a landing airplane occupies the runway while another one waits to take-off. In the second half, the just-landed airplane stays in the parking lot while the other airplane takes off.

Simulating both scenarios with $k = 10\ min$ yielded the expected results.

5.2.2 Exponential case

We studied the model under exponentially distributed times. Designing a Queue Theory model for the system proved to be difficult, since the system is made of one runway that serves two queues with different priorities and with non-independent arrivals. We decided to model an approximated version of the system, where the two queues are independent.

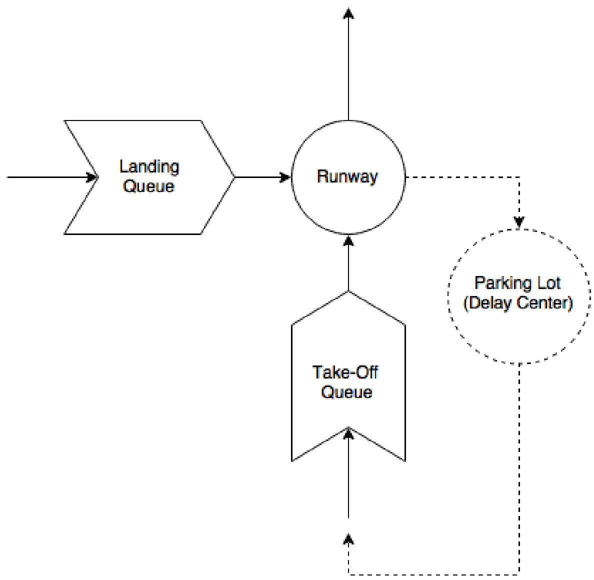

FIGURE 3 - QUEUE THEORY MODEL

[Figure 3] shows both the simplified model and the real model. Dotted lines are used for the elements that were simplified out. In the simplified model, we can set the arrival rate to λ for both queues; this is justified in the real model because, as long as the system is stable, the mean arrival rate to both the parking lot and the takeoff queue will be equal to λ.

The simplified and real models will behave the same for the landing queue. In fact, even if the rest of the models differ, the priority placed on the landing airplanes will hide those effects.

For the take-off queue, instead, there is a non-trivial change: the topology of the real model introduces a correlation between the two arrival processes, that will in turn introduce a variation in waiting times that we cannot predict by means of the simplified model. Although, we expect the simplified model to be reliable for low utilizations, since the runway will be empty for most of the time reducing the effects introduced by the correlation.

We define λ as the arrival rate of airplanes of both queues; μ_l as the service rate for landing airplanes; μ_t as the service rate for taking-off airplanes. T_l and T_t are the response times, N_l and N_t are the number of airplanes in the queues. $\rho_i = \frac{\lambda}{\mu_i}$ is the utilization for each priority queue. The mean response times and mean number of airplanes in the queues are described by the following equations:

$$E[T_l] = \frac{\rho_t \mu_l + \mu_t}{(1-\rho_l)\mu_l \mu_t}$$

$$E[T_t] = \frac{\rho_l E[T_l]\mu_t + 1}{(1-\rho_t)\mu_t}$$

$$E[N_l] = \lambda E[T_l]$$

$$E[N_t] = \lambda E[T_t]$$

We set t_l, t_t, t_p and varied λ for every scenario, so to have different utilizations. The data confirmed that the simulation software follows the approximate model for low utilizations on the takeoff queue, while on the landing queue yields the expected results for all values of utilization.

6. Study #1: queue lengths and waiting times on varying workloads

For this study, we set the service times of the airport to reasonable values while using the interarrival time to issue different workloads to the system. We measure the workload as the utilization of the runway, that is the sum of the utilization of landing and take-off queue. We set the mean times as $t_l = 5\,minutes$, $t_t = 5\,minutes$ and $t_p = 45\,minutes$. Given a target workload of $\rho = \rho_i + \rho_t = \lambda t_l + \lambda t_t$ we can derive the required λ, which is the inverse of the interarrival time. Every experiment has been repeated 50 times in independent conditions and was simulated for 25 days, including the warmup time. We set the study using three different sets of distribution:

- Deterministic values for all factors
- Exponential distribution for all factors
- Log-normal distribution for service times and exponential interarrival times

In all three cases, we used the mean values mentioned above, while in the lognormal case we set the standard deviation to 10% of the mean, so that most of the values fall within 30% from the mean. The third case is the closest to reality, with a memoryless arrival process while the time required for landing, takeoff or parking is (almost) always close to the mean value.

We tried fitting waiting times to theoretical distributions. We know from Queue Theory that in a simple service center the waiting times are 0 with a probability $(1-\rho)$ and exponentially distributed in the rest of the cases. Even in our model, the waiting times must be 0 with a probability $(1-\rho)$ where $\rho = \rho_l + \rho_t$; therefore, we fitted the sample excluding null values; we can get the whole distribution through total probability theorem.

Another insight we sought, for exponential and lognormal cases, is the relation between the workload and the dimensioning required for the airplanes and the airport. In fact, airplanes arriving at the airport should carry an excess amount of fuel needed to stay in air while waiting to land; the airport should have enough space for the airplanes to stay while waiting for take-off. We decided to value these dimensioning parameters as the 0.9 quantiles of the waiting time on the landing queue and the size of the queue for the takeoff queue. Using these values will ensure that there will be enough space or fuel to withstand the requests in 90 % of the cases. We assume that, for a physical implementation of the system, emergency policies will be in place that may change the priorities to deal, for example, with an airplane that is ending its fuel due to an excessive waiting time.

To compute the 0.9 quantile for a utilization, we extracted the 0.9 quantile from every independent sample i obtained from that factors set, generating a sample of 50 IID RV $X_1, X_2, \dots X_{50}$. We computed the sample mean, which is a Normal RV by the CLT, and computed 95% level CI. This was repeated for every utilization. We computed a regression model for the quantiles of take-off queue length to predict a reasonable size and for the quantiles of landing queue waiting time to predict the fuel reserve of landing airplanes for any given ρ.

6.1 Deterministic study

Let's assign to every airplane a natural number that represents the order in which they arrive at the airport. Now, if we consider the intervals of time in which the runway is occupied by a landing airplane or by a taking-off airplane we can obtain respectively:

- $[\, nt_i \,;\, nt_i + t_l \,)$
- $[\, mt_i + t_l + t_p \,;\, mt_i + t_l + t_p + t_t \,)$

where $n \in \mathbb{N}^+$ and $m = n - k$, with $n > k,\ k \in \mathbb{N}^+$. If we want no waiting times in both queues we must ensure that the two intervals are disjoint for each n, m. With some computations, we can obtain the following condition:

$$\frac{t_p}{t_i} + \rho \leq\ k \leq \frac{t_p}{t_i} + 1$$

If a k exists that verifies this condition then the waiting times will be always 0 for every airplane, both in the landing queue and in the takeoff queue. Another form for this condition is this:

- (1). $k \cdot t_i \geq t_l + t_p + t_t$
- (2). $(k-1) \cdot t_i \leq t_p$

We can understand what k means from these two conditions. For every airplane $n,\ m = n - k$ is the id such that every airplane $i \leq m$ has already left the airport and every airplane $i > m$ is still inside the parking lot. Therefore, there is no airplane other than n trying to occupy the runway which means no queueing time.

6.2 Study with exponential interarrival and service times

6.2.1 Waiting time study

We have analyzed the PDFs of waiting times in both queues with several values of ρ. Since all the times are exponentially distributed, we tried to fit an exponential distribution, getting a very good fit for the landing queue (Figure 4), and a good fit for the take-off queue (Figure 6). The latter shows a much shorter tail than the exponential distribution, especially for larger ρ (Figure 7). We believe this is caused by physical constraints of the system. The figures presented represent these behaviors for $\rho = 0.8$.

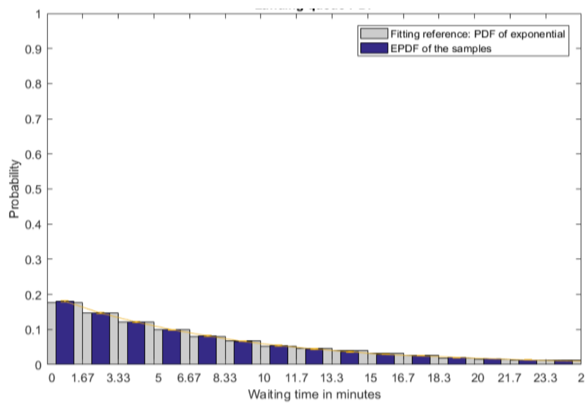

FIGURE 4 - LANDING QUEUE WAITING TIMES PDF

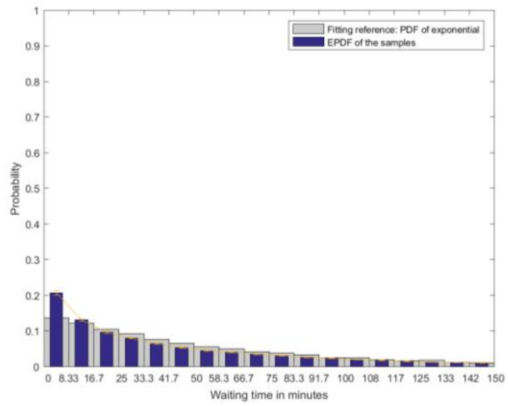

FIGURE 6 - TAKEOFF QUEUE WAITING TIMES PDF

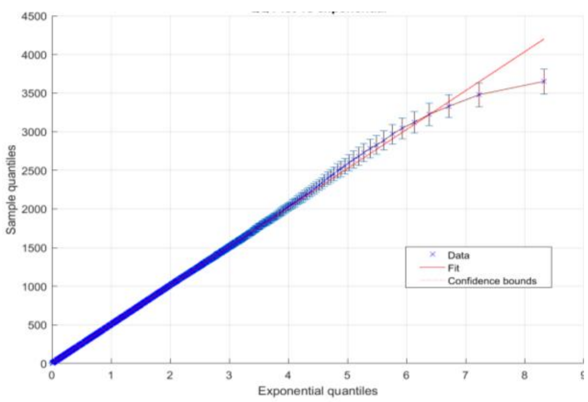

FIGURE 5 - QQ PLOT FOR LANDING QUEUE WAITING TIMES VS EXPONENTIAL

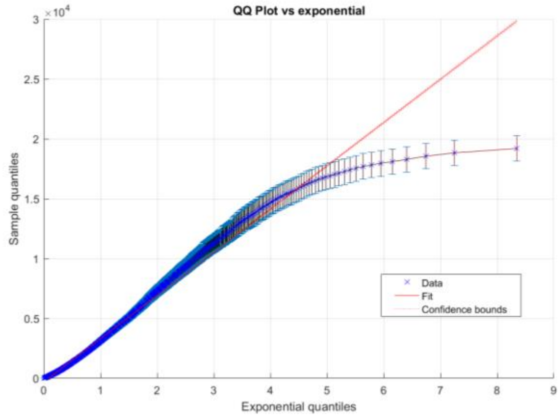

FIGURE 7 - QQ PLOT FOR TAKEOFF QUEUE WAITING TIMES VS EXPONENTIAL

The sample mean of waiting time vs. runway utilization shows a linear behavior in landing queue. The mean waiting time of the takeoff queue shows instead an exponential behavior, that we studied through a logarithmic transformation. Using linear fitting on the transformed data we found the coefficient for $log(y) = b_0 + b_1 x$ as $b_0 = 3.155, b_1 = 5.966$ with 95% CI of, respectively, 0.1526 and 0.2740. Anti-transforming the obtained regression we obtain the exponential regression in figure.

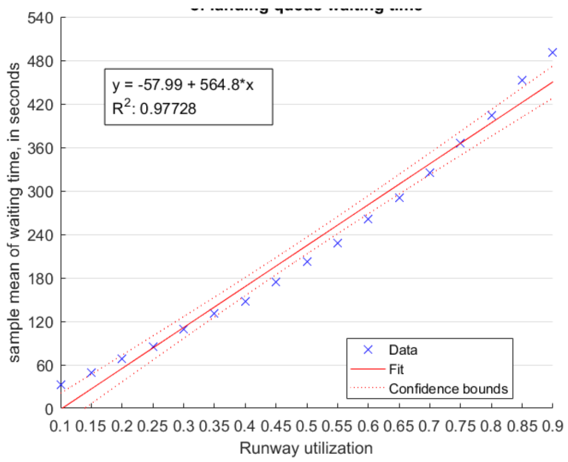

FIGURE 8 - LINEAR REGRESSION FOR SAMPLE MEAN OF LANDING QUEUE WAITING TIME

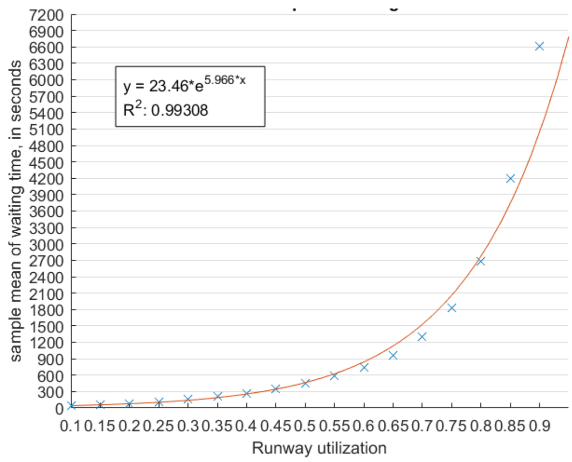

FIGURE 9 - EXPONENTIAL REGRESSION FOR SAMPLE MEAN OF TAKEOFF QUEUE WAITING TIME

6.2.2 Dimensioning study

Studying the waiting times at the landing queue, we found that the 0.9 quantile increases linearly with the runway utilization, as shown in [Figure 10] going from ~2 minutes at 15% utilization to ~20 minutes at 90%. Moving to lower utilization the decrease will be non-linear, as shown by the value at 10%, because the waiting time rapidly goes to 0 as fewer airplanes arrive at the airport.

The length of the takeoff queue has instead a more complex behavior. The 0.9 quantiles are found to be 0 for utilization up to 35 %. Between 35% and 45% it shows a linear behavior that we chose not to investigate as

1) we would need more data, which means another set of simulations focusing on that interval
2) being the values in this interval lower than 1, it would be rather useless to know about the exact values, as the recommended dimensioning would remain 1 within this interval anyway.

For utilization of 45% and higher the quantiles show an exponential behavior. Following the same procedure as for the sample mean waiting time, we found transformed linear parameters $b_0 = -4.399, b_1 = 8.469$ with 95% CI of, respectively, 0.1934 and 0.2802. By anti-transforming we obtained the exponential regression shown in [Figure 11].

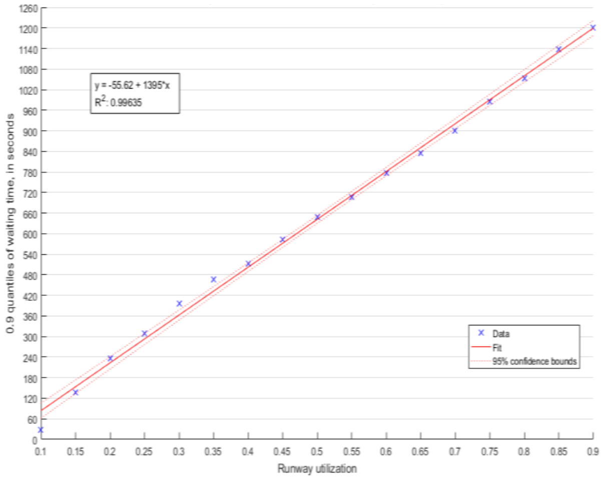

FIGURE 10 - LINEAR REGRESSION FOR 0.9 QUANTILES OF LANDING QUEUE WAITING TIME

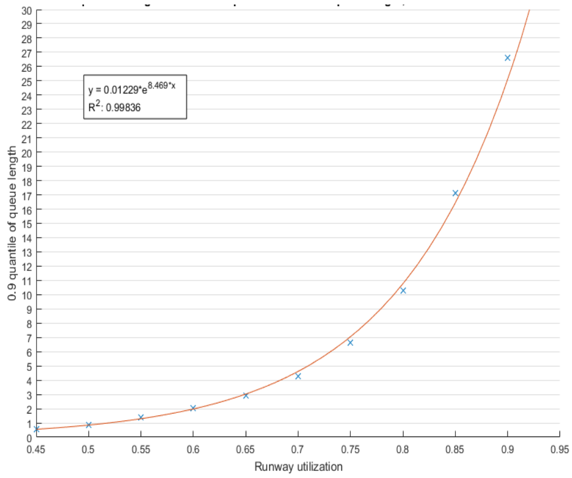

FIGURE 11 - EXPONENTIAL REGRESSION FOR 0.9 QUANTILES OF TAKEOFF QUEUE LENGTH, FOR UTILIZATION OVER 45%

6.3 Study with log-normal distributed times and exponential interarrival times

The waiting times show the same behaviors as the previous study, but with smaller values. The values of their quantiles are a little more than half of the quantiles for the previous study.

6.3.1 Waiting Times

We tried fitting the waiting times to several theoretical distributions. The most interesting ones were the uniform distribution and the exponential distribution. We noted, through QQ-plots, that the fitting to a uniform distribution was almost perfect except for the tail, which was longer in the sample. Instead, the exponential showed to be a good fit for the tail, except for the far end of the tail, which was shorter in the sample. We also noticed, through histograms, that the uniform was a good fit for values in $(0, 5min] = (0, t_l]$, and after that it had a completely different behavior. We tried fitting the values after t_l with an exponential. The histograms, as shown in [Figures 12-15] confirmed our hypothesis.

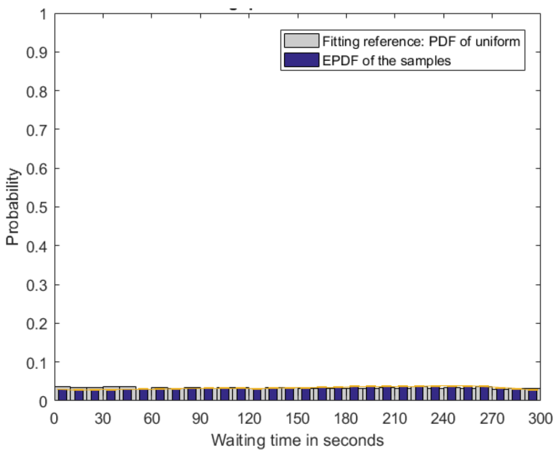

Figure 12 - PDF of waiting times in landing queue from 0 to 5 minutes

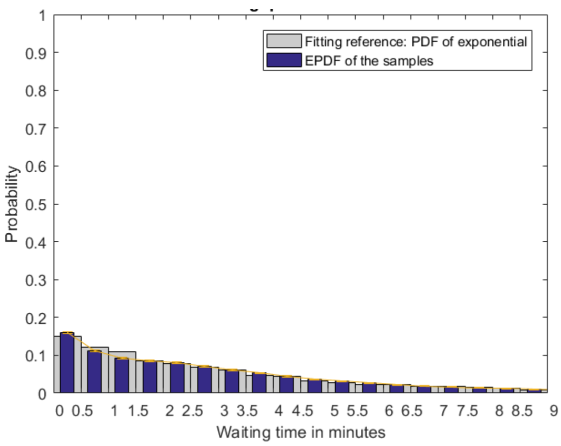

Figure 13 - PDF of waiting times in landing queue from 5 minutes

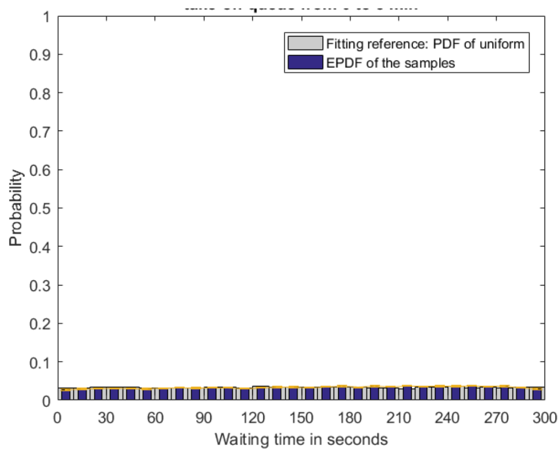

Figure 14 - PDF of waiting times in takeoff queue from 0 to 5 minutes

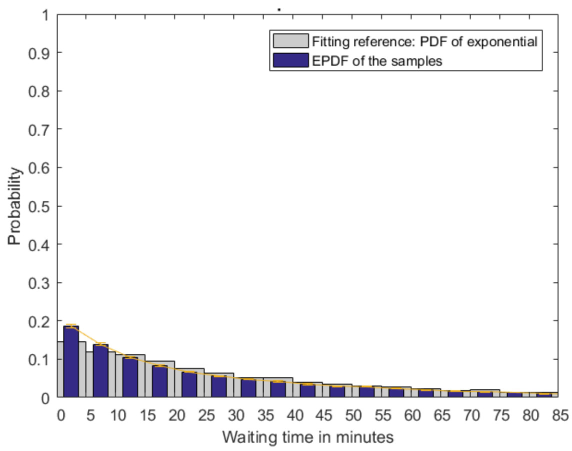

Figure 15 - PDF of waiting times in takeoff queue from 5 minutes

We can explain this behavior if we consider that the lognormal distribution is not memoryless. When an airplane enqueues, there are three cases:

- The runway is empty, and so its waiting time is null;
- The runway is busy but the queue is empty;
- The runway is busy and the queue is not empty;

We already considered the first case, so it's not interesting. The second case is what causes the presence of a uniform behavior: the enqueuing airplane will wait the residual service time of a lognormal distribution, which will not be distributed like a lognormal at all. Since our lognormal has low variance, we can assume it to be almost always equal to the mean service time. There is no preference at which point of the service time the airplane will enqueue, thus we can describe the time that has passed since the start of the service as a uniform RV, let's say $a \in U(0, t_l)$. It easily follows that the residual service time is $t_l - a$, which is a uniform RV with the same range as a. The third case is what causes the exponential tail.

It is to note that we have a simple uniform RV because $t_l = t_t$ and so $\rho_l = \rho_t$. If they were different, we wouldn't have a uniform, but a sum of uniforms with different ranges, scaled by the respective ρ, for total probability.

The sample mean of waiting time vs. runway utilization shows a linear behavior in landing queue, as shown in [Figure 16]. The same shows an exponential behavior in take-off queue. Following the same procedure as for the exponential case, we found transformed linear parameters $b_0 = 2.264, b_1 = 6.295$ with 95% CI of, respectively, 0.1888 and 0.3392. By anti-transforming we obtained the exponential regression in [Figure 17].

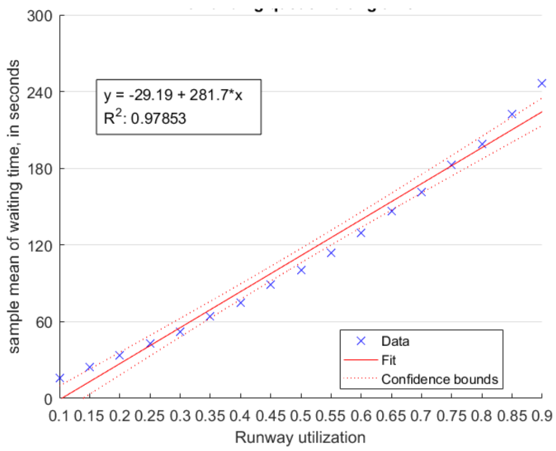

FIGURE 16 - LINEAR REGRESSION FOR SAMPLE MEAN OF LANDING QUEUE WAITING TIME

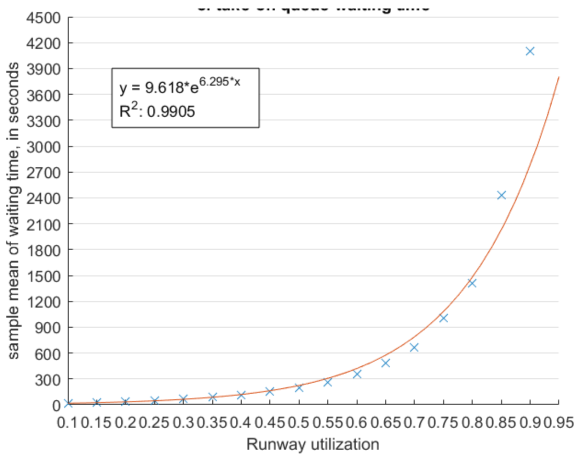

FIGURE 17 - EXPONENTIAL REGRESSION FOR SAMPLE MEAN OF TAKE-OFF QUEUE WAITING TIME

6.3.2 Dimensioning study

Studying the quantiles in the lognormal case, we found a similar behavior with respect to the exponential case, obviously with different values. For the waiting times at the landing queue, we found that the 0.9 quantile increases linearly with the runway utilization, as shown in [Figure 18] going from less than 2 minutes at 15% utilization to ~9 minutes at 90%. Again, moving to lower utilization the decrease is non-linear because the waiting time rapidly goes to 0 as fewer airplanes arrive at the airport. The quantiles of the length of the takeoff queue is split in two sections. Up to 45% utilization the queue is empty, while from 50% on it increases exponentially. Following the same procedure as for the exponential case, we found transformed linear parameters $b_0 = -6.4206, b_1 = 10.234$ with 95% CI of, respectively, 0.4173 and 0.5863. By anti-transforming we obtained the exponential regression in [Figure 19].

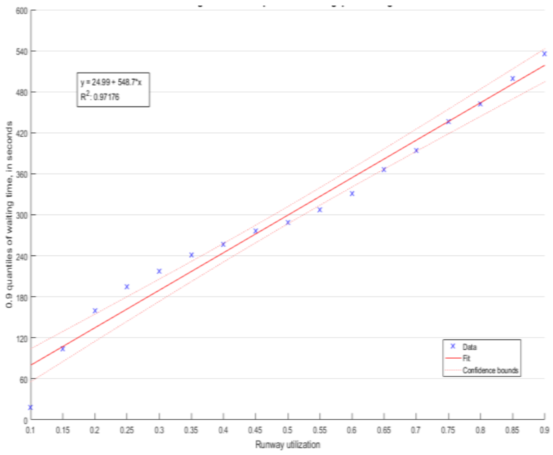

FIGURE 18 - LINEAR REGRESSION FOR 0.9 QUANTILES OF LANDING QUEUE WAITING TIME

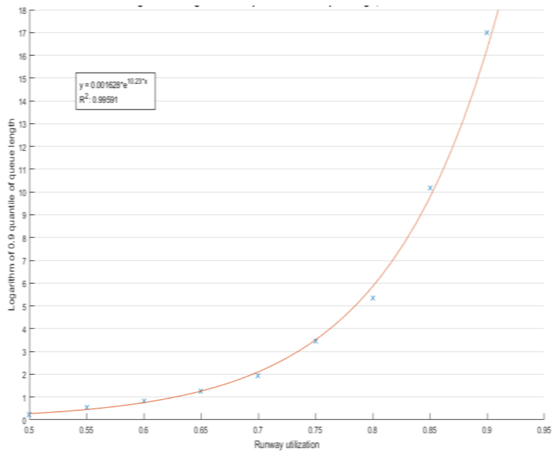

FIGURE 19 - LINEAR REGRESSION FOR LOGARITHM OF 0.9 QUANTILES OF TAKEOFF QUEUE LENGTH, FOR UTILIZATION OVER 50%

6.4 Waiting times fairness

We studied the fairness of queue waiting times with service times extracted from an exponential or a lognormal distribution, both with exponential interarrival times. We studied also the case where both interarrival and service times are deterministic. The latter proved to be less interesting, since the fairness depends greatly on how the times interleave; a properly chosen scenario can have high utilization and maximum fairness, another one can have same utilization and lower fairness. In the cases where service times are RV, when ρ is higher, so is fairness. This is due to the fact that there is a $(1-\rho)$ probability that the queue is empty and the waiting time will be 0. When ρ is low, almost every airplane will have null waiting time, but some of them will have possibly very high waiting times, thus causing high unfairness. On the other hand, when ρ is high, almost no one will have null waiting times, and the fairness will depend on the distribution of the waiting times conditioned to non-null values.

Given the same ρ, for log-normal service times, waiting times are a little fairer than in the exponential service times case; the effect is easier to notice at high utilizations, where for the same reasons above, the fairness highly depends on the distribution of waiting times; in both cases service times have the same mean; our log-normal distributions have a greatly smaller variance with respect to exponential distribution since we set a low σ, thus they are more fair. We believe this influences queue waiting times, making them more fair in the log-normal service times cases.

We can compute waiting time as $W_{qr} = \sum_{i=1}^{N_q} T_{qi} + T_r$, where N_q is the number of airplanes in the queue, T_{qi} is the service time RV of enqueued airplane i and T_r is the residual service time RV. Its expected value is $E[W] = n_q \cdot t_q + t_r$.

If we exclude the residual service time, we can say that $W_q = \sum_{i=1}^{N_q} T_{qi}$ and $W_{qr} = W_q + T_r$. W_q fairness depends mainly on the service time fairness (hence on its variance). Thus, the lognormal distribution, which has a greatly smaller variance than the exponential distribution, makes W_q greatly fairer. T_r softens this effect, since it is very different from one case to the other. With exponential service times, T_r is distributed identically to service times, being exponential memoryless; when service times are lognormal, T_r is a uniform RV $U(0, T_{q0})$, as we noted before. This time, the variance of the exponential residual time is far smaller than the uniform one, compensating the effect of the lognormal in W_q. This has an intuitive explanation: in the lognormal case the time at which the airplane enqueues affects the waiting time it will experience. If it enqueues when the runway has just started servicing, it will wait a whole service time, almost always equal to t_q. If it does when the runway has almost ended, it will have null residual service time. This cases are equally probable, thus fairness is reduced.

The reason why higher values of ρ make this effect more noticeable can be explained with few algebraic computations. As we said before $W_{qr} = \sum_{i=1}^{N_q} T_{qi} + T_r$, which becomes $W_{qr}^e = \sum_{i=1}^{N_q+1} T_{qi}$ for the exponential since $T_r \sim T_q$; $W_{qr}^l \in \left[\sum_{i=1}^{N_q} T_{qi}, \sum_{i=1}^{N_q+1} T_{qi}\right]$ for lognormal since $T_r \sim U(0, T_{q0})$. At high values of ρ we have that $N_q \gg 1$, therefore $W_{qr}^e \approx W_{qr}^l \approx \sum_{i=1}^{N_q} T_{qi}$, which means that fairness will depend mainly on the service time distribution fairness.

In the exponential service times cases, landing and takeoff queue are equally fair. On the other hand, in the log-normal service times cases, landing queue is fairer than the take-off one. We think that this is caused by priority scheduling combined with the absence of memoryless property. The priority policy makes some airplanes in the take-off queue wait more than others: if a landing airplane arrives, it will take precedence over the take-off airplanes, thus all airplanes in queue will wait one service time more than the ones that get served without being interrupted. As ρ gets higher, this happens more often and becomes easily noticeable.

7. Study #2: parking occupancy on varying workloads

We decided to study the parking occupancy with a factorial design approach. We set the following extremes for every factor:

- $t_l \in [5, 15]\ min$
- $t_p \in [30, 60] min$

We set $t_i = 30\ min$ and $t_t = 5\ min$ for every scenario. Varying t_l, ρ also changes. We saw earlier that for different values of ρ the system has a different behavior. This however influences only the queues that use the runway, while the average flow of incoming airplanes remains the same as t_i remains constant. Thus, the parking occupancy is not influenced by the varying ρ. Every scenario was repeated 50 times under independent initial conditions and was simulated for 25 days. We set this study using three different sets of distribution:

- Deterministic times for all factors
- Exponential distribution for all factors
- Log-normal distribution for service times and exponential interarrival times

In all three cases, we used the mean values mentioned above, while in the lognormal case we set the standard deviation to 10% of the mean, so that most of the values fall within 30% from the mean. The deterministic workload was studied analytically and an equation was found for the parking lot size.

7.1 Deterministic study

For a deterministic workload, we decided to set the size of the parking lot as the maximum number of airplanes throughout the whole measurement.

In a stable state the rate at which airplanes enter the system is the same at which airplanes enter and exit the parking lot. Therefore, the parking alternates an entering and a leaving airplane. As such, if we compute the number of airplanes that enter the parking lot before the first one leaves we will have the maximum size of the airport.

The number of airplanes that managed to enter before the first left can be computed as $S = \left\lceil \frac{t_p}{t_i} \right\rceil$. After that, the parking lot occupancy will oscillate between $S - 1$ and S. Thus, S is the maximum number of airplanes that the parking lot will have to contain at the same time, and we should size it according to S.

7.2 Study with exponential interarrival and service times

In this case, we cannot reliably measure the maximum occupancy of the parking lot, so we settled at measuring the 0.9 quantile of it. We did a 2^kr study with the quantiles collected in every scenario, to assess the relation between the factors t_l, t_p and the parking lot occupancy.

Once obtained the data, we tried with a logarithmic transformation obtaining a fairly constant standard deviation for residuals, that are also distributed like IID RVs. We will call q_0 the mean value between all scenarios, q_l the coefficient relative to t_l, q_p the one relative to t_p, q_{lp} the one relative to their interaction. S will be the size of the parking lot and it will be:

$$S = e^{q_0} \cdot e^{q_l x_l} \cdot e^{q_p x_p} \cdot e^{q_{lp} x_{lp}}$$

We obtained the following results:

- $q_0 = 0.9415; q_l = 0.0025; q_p = 0.3024; q_{lp} = 8.6104 \cdot 10^{-4}$
- $f_l \simeq 0\%; f_p = 98.42\%; f_{lp} \simeq 0\%; f_e = 1.57\%$
- $CI \ = \ 0.0054$ (95% level of confidence) for every q

The size of the parking lot is completely due to the parking time, while interactions and errors are negligible. This can be explained thinking back to what we found in the deterministic study, where we linked the maximum parking lot occupancy to $\left\lceil \frac{t_p}{t_i} \right\rceil$ which is independent from t_l.

Even if t_l changes, the number of airplanes that manage to get in the parking lot before an airplane comes out of it is determined only by the ratio $\frac{t_p}{t_i}$, while t_l may only delay the flux of airplanes of a random amount, which has almost no influence. This result is, moreover, independent from the steadiness of the system, as the priority on the landing queue assures to propagate the mean arrival rate to the parking lot.

We decided to try a second set of experiments to check if parking size was really influenced by t_i even if times were RVs. So, we set $t_l = 5min$ and $t_i \in [15,30]min$. We used logarithmic transformation and the hypotheses were verified. The results are as we predicted:

- $q_0 = 1.2361; q_i = -0.2970; q_p = 0.2954; q_{ip} = 0.0062$
- $f_i = 49.93\%; f_p = 49.40\%; f_{ip} \simeq 0\%; f_e = 0.64\%$
- $CI = 0.0048$ (95 % level of confidence) for every q

From this results we can see that the landing time and parking time have almost the same influence; a larger t_p makes the size grow larger, a larger t_l makes it smaller. This makes sense if we consider the maximum size for the deterministic case, that has t_p at numerator and t_i at denominator. The interaction between times is negligible; errors are negligible too.

The relationship found in the deterministic case may also explain the need for a logarithmic transform, as $\log\left(\frac{t_p}{t_i}\right) = \log(t_p) - \log(t_i)$ which is a linear relation with equal weights but opposed signs, which is confirmed by the results of the both 2^kr studies: in the first, being t_i constant we only measured the positive influence of t_p; while in the second study we measured both with equal influence but opposed signs.

From Queue Theory, we know that the distribution of the size of the parking lot, which is a delay center, is a Poisson distribution with mean $\frac{\lambda_p}{\mu_p}$. We can now safely say that this mean is equal to $\frac{t_p}{t_i}$.

7.3 Study with exponential interarrival times and lognormal distributed service times

We tried the same sets of experiments with lognormal distributed times. The results didn't change much, except that we have smaller values for the parking lot size. We applied a logarithmic transform and verified the hypotheses.

- $q_0 = 0.8862; q_l = -0.0449; q_p = 0.3095; q_{lp} = 0.0077$
- $f_l = 2.05\%; f_p = 97.29\%; f_{lp} \simeq 0\%; f_e = 0.6\%$
- $CI = 0.0034$ (95% level of confidence) for every q

Landing time is still not influent on the parking lot size. We tried also the experiment where we change t_i, obtaining again the same results. Again, we applied the logarithmic transform and verified the hypotheses.

- $q_0 = 1.2203; q_i = -0.2891; q_p = 0.2984; q_{lp} = 0.0033$
- $f_i = 48.26\%; q_p = 51.40\%; f_{ip} \simeq 0\%; f_e = 0.33\%$
- $CI = 0.0034$ (95% level of confidence) for every q

Summing up, all these experiments served to prove that t_l has little if any influence on the parking occupancy, while t_i and t_p have equal influence on it, and that this is a property of the system itself and is not caused by properties of the distributions of interarrival and service times.

8. Software tools used

For the software implementation, we used the OMNeT++ framework in C++. We used the graphing tools included in the IDE to study the warmup time. For other studies, we used the gawk text processing language to parse the framework result files in .csv format and .m MATLAB script files. The PowerShell scripting language was used to prepare the simulation scenarios, simulate them and call gawk scripts on their results. Studies on the results were then carried on MATLAB. The scripting code setup used, along the experiments, is included with the simulator source code under the *simulation*s directory.