# Analysis of Air Transportation Systems

## Modeling and Simulation and its Application to Capacity and Delay Estimation

**Dr. Antonio A. Trani**

**Professor of Civil and Environmental Engineering Virginia Polytechnic Institute and State University**

**Blacksburg, Virginia**

**Spring 2018**

# Outline of this Presentation

- Review of simulation and modeling techniques
  - Analytic Models (Prelude to Simulation)
  - Monte Carlo Simulation
  - Continuous Simulation
  - Discrete Simulation
- Future directions in airport simulation
  - Object oriented modeling and simulation
  - Distributed simulation and visualization techniques

# Introduction to Simulation

A definition of simulation is:

- A technique used to predict the behavior of complex systems over time
- Simulation entails the use of a computer to evaluate models numerically (Law and Kelton, 1991)

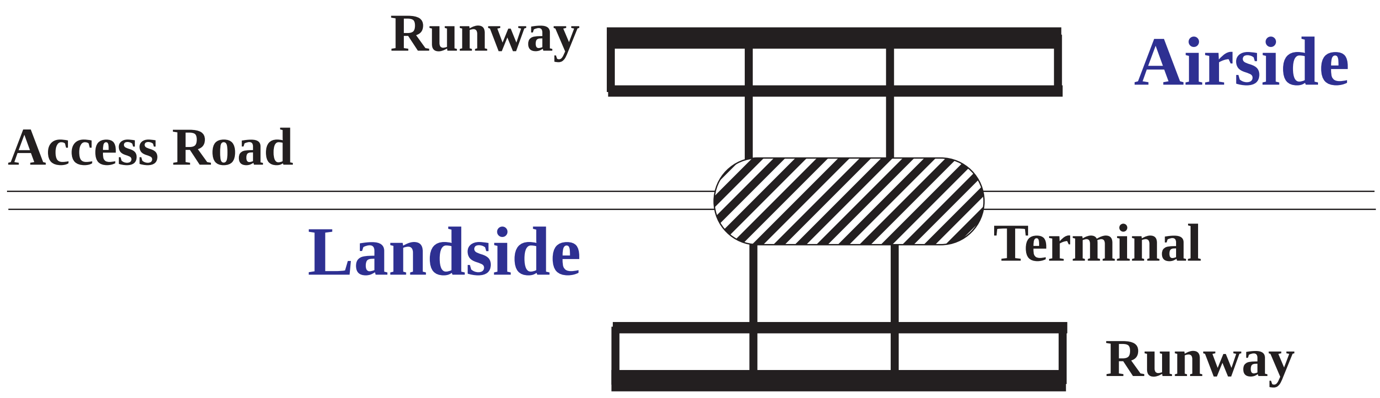
Simulation is heavily used among operations research analysts ranking in second place after linear programming (Hillier and Lieberman, 1996)

Simulation is part art and part science

# Definition of System

A system is a collection of elements that act and interact towards a common goal

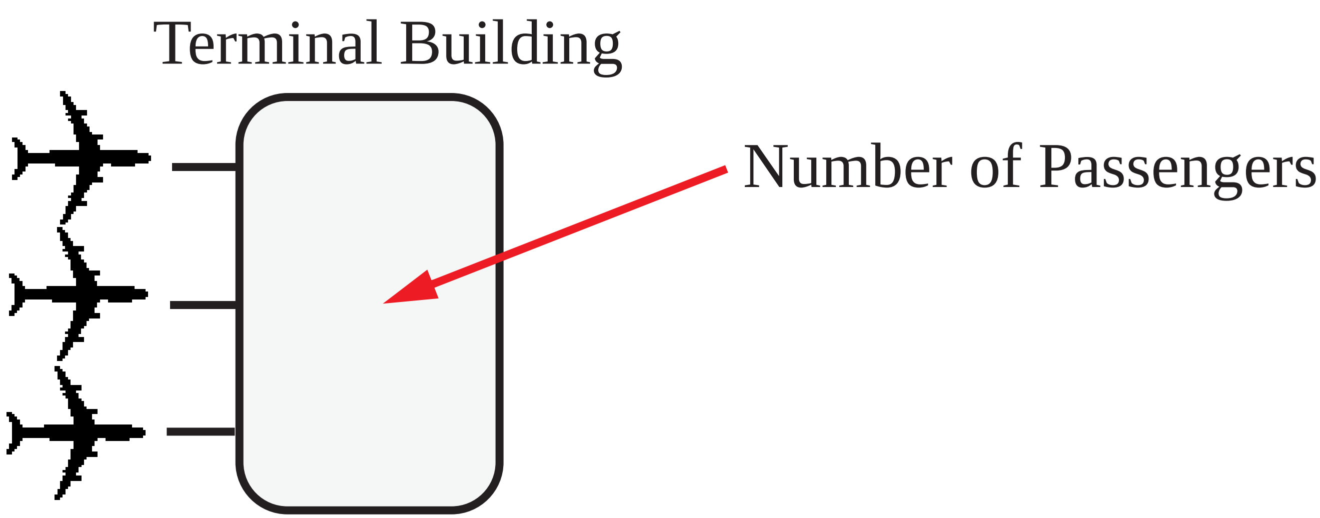
- Airports have airside and landside elements



# State of a System

The state of the system is defined by a set of variables describing the system at some point in time

- The state of an airport terminal can be dictated by the passenger flows traversing the terminal over time

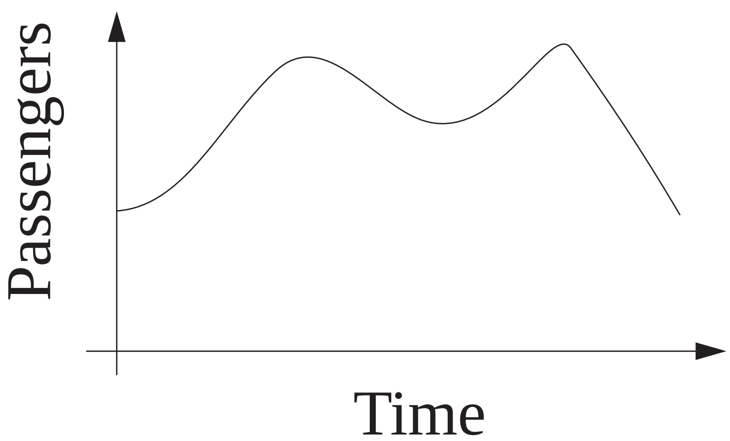


# Airport Simulation

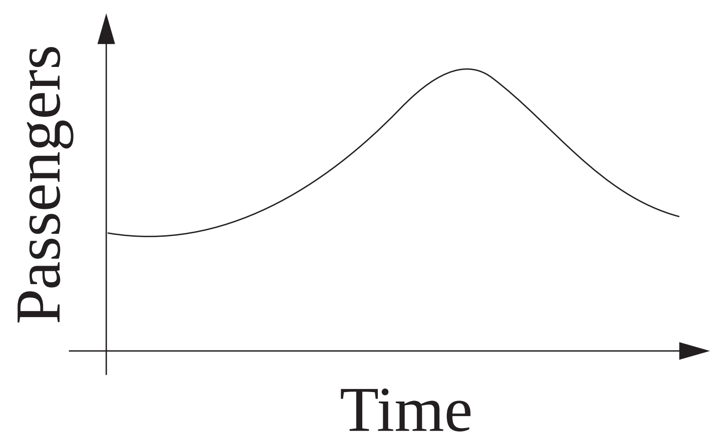
Simulations are carried out using models

- We could use simulation to predict passenger flows inside each element of an airport terminal

Ticket Counter Area




Security Area

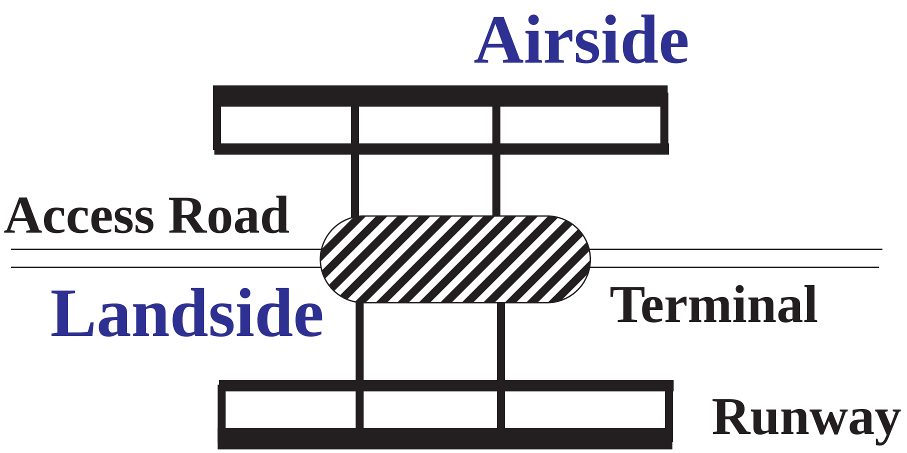


# Airport Modeling and Simulation Domains

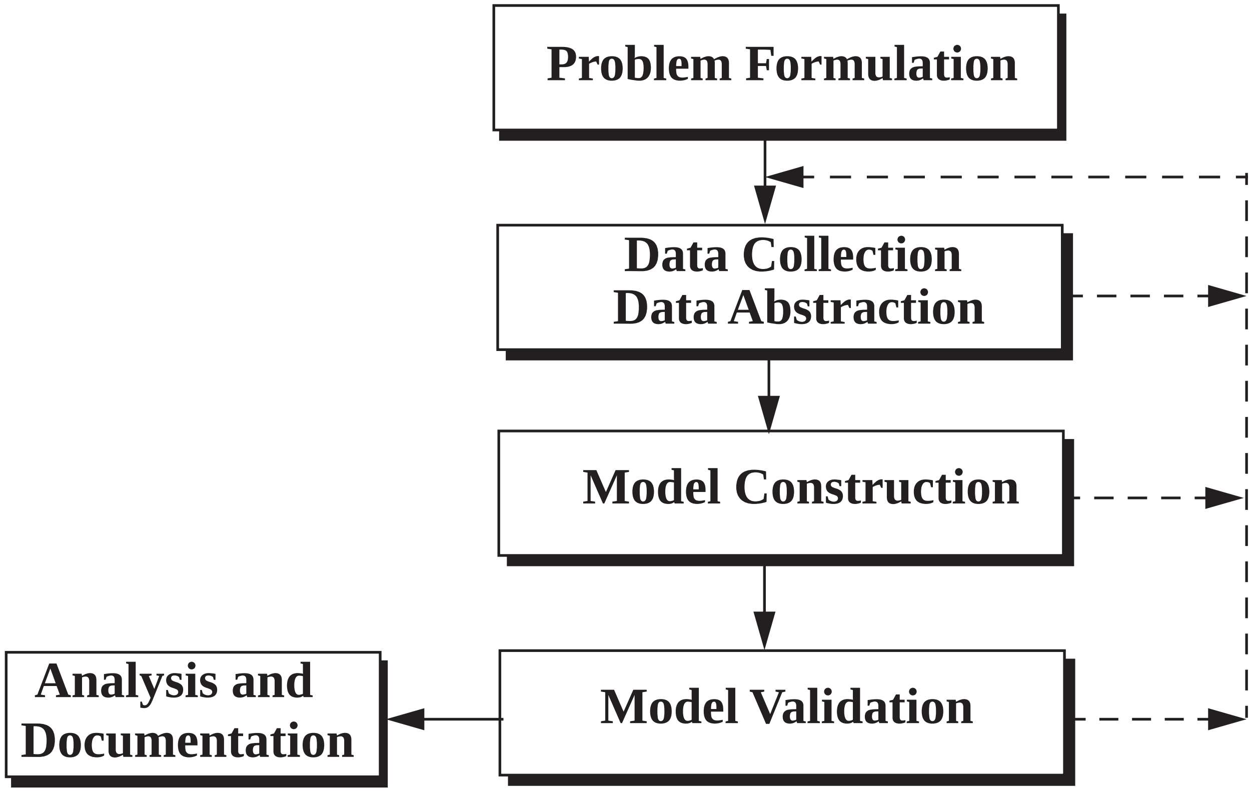
Traditionally airport models and simulations have addressed distinct components independently

- This has been done to simplify the analysis of three distinct flows in the airport system

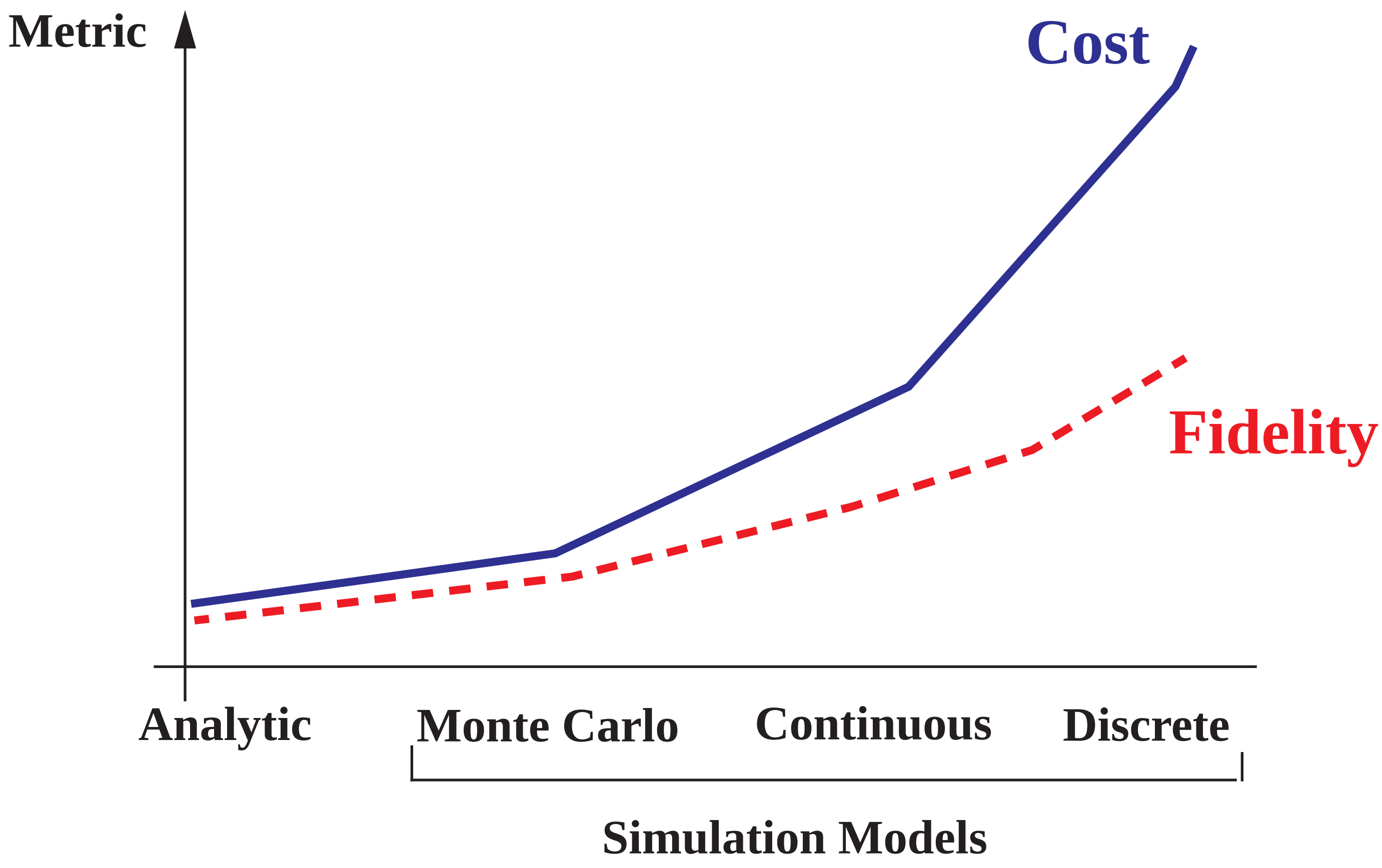
| Analysis | Model |
|---|---|
| Airside | SIMMOD |
| Landside (Terminal) | ALSIM |
| Landside (Access) | Integration |



# Modeling and Simulation Process



# Simulation Effort and Cost



# Modeling and Simulation Techniques

- Analytic Models

  - Exact representations of the system (close form solution)

- Monte Carlo Simulation

  - Description of a system with random variables without consideration of the passage of time

- Continuous Simulation

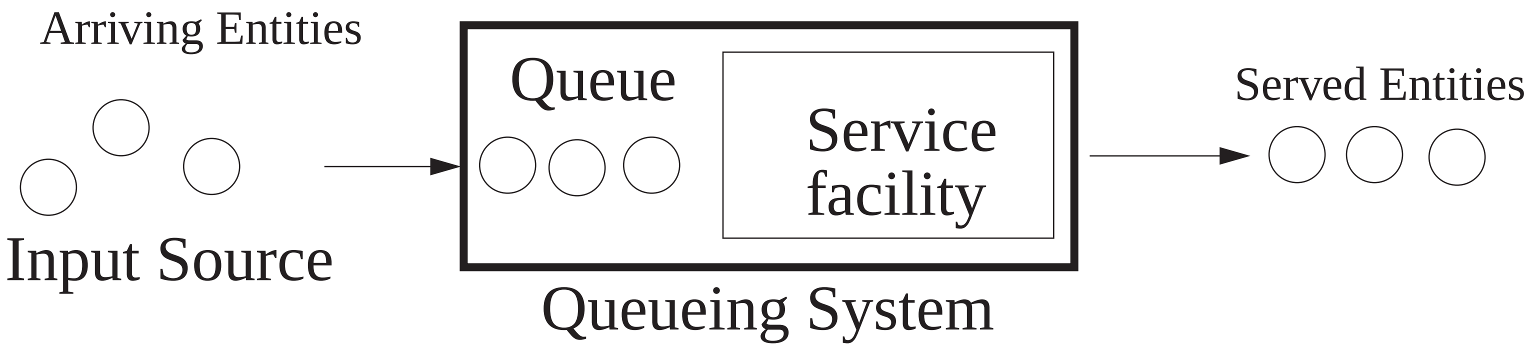
  - Description of a system using differential equations detailing how state variables change over time

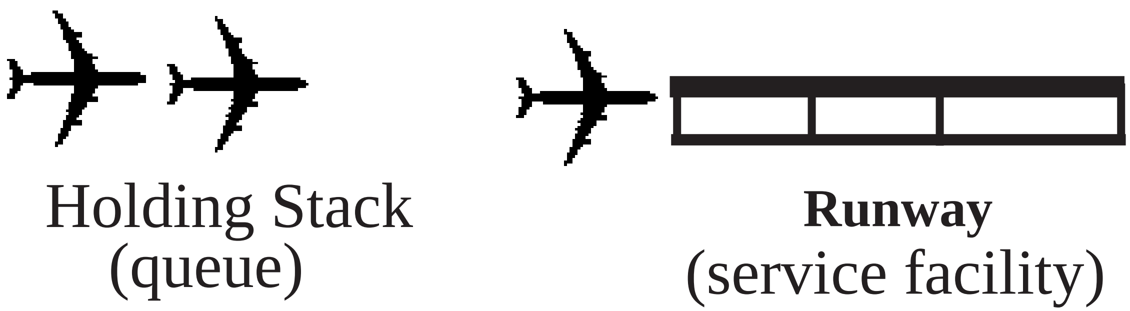
- Discrete Event Simulation

  - Description of a system using logical relationships detailing how state variables change over time (discrete changes)

# Analytic Model Example (I)

An example of an analytic model is a close form queueing model to estimate airside capacity and delay



# Analytic Model Example (II)

Specification of a Queue

- Size of input source
- Input function
- Queue discipline
- Service discipline
- Service facility configuration
- Output function (distribution of service times)

# Analytic Model Example - Nomenclature

| Parameter | Meaning |
|---|---|
| $\lambda$ and $\mu$ | Arrival and service rates (entities/time) |
| $s$ | Number of servers |
| $L$ | Expected number of entities in system |
| $P_n$ | Probability of having $n$ in the system |
| $\rho$ | Utilization factor of facility |
| $W_q$ | Average waiting time in the queue |
| $W$ | Average waiting time in the system |

# Basic Multiserver Queueing Equations

Assume an infinite source queue with constant $\lambda$ and $\mu$

- Poisson arrivals with parameter $\lambda_n$
- Probability function of service completions is negative exponential with parameter $\mu_n$
- Only one arrival or service occurs at a given transition

For more information on queueing models consult any Operations Research textbook (i.e., Hillier and Lieberman, 1996)

# Multi-server Queueing Equations (I)

$\rho = \lambda / s\mu$ utilization factor

Probabilities of zero and $n$ entities in the system

$$P_0 = 1 \Big/ \left( \sum_{n=0}^{s-1} \frac{(\lambda/\mu)^n}{n!} + \frac{(\lambda/\mu)^s}{s!} \left( \frac{1}{1 - (\lambda / s\mu)} \right) \right) \qquad (1)$$

$$P_n = \begin{cases} \dfrac{(\lambda/\mu)^n}{n!} P_0 & 0 \le n \le s \\[2ex] \dfrac{(\lambda/\mu)^n}{s! s^{n-s}} P_0 & n \ge s \end{cases} \qquad (2)$$

# Multi-server Queueing Equations (II)

Expected no. of entities in system

$$L = \frac{\rho P_0 \left(\frac{\lambda}{\mu}\right)^s}{s!(1-\rho)^2} + \frac{\lambda}{\mu} \qquad (3)$$

Expected no. of entities in queue

$$L_q = \frac{\rho P_0 \left(\frac{\lambda}{\mu}\right)^s}{s!(1-\rho)^2} \qquad (4)$$

# Multi-server Queueing Equations (III)

Average waiting time in queue

$$W_q = \frac{L_q}{\lambda} \qquad (5)$$

Average waiting time in system

$$W = \frac{L}{\lambda} = W_q + \frac{1}{\lambda} \qquad (6)$$

# Multi-server Queueing Equations (III)

Waiting time probability

$$P(W > t) = e^{-\mu t}\left[1 + \frac{P_0\left(\frac{\lambda}{\mu}\right)^s}{s!(1-\rho)}\left(\frac{1 - e^{-\mu t(s - 1 - \lambda/\mu)}}{s - 1 - \lambda/\mu}\right)\right] \quad (7)$$

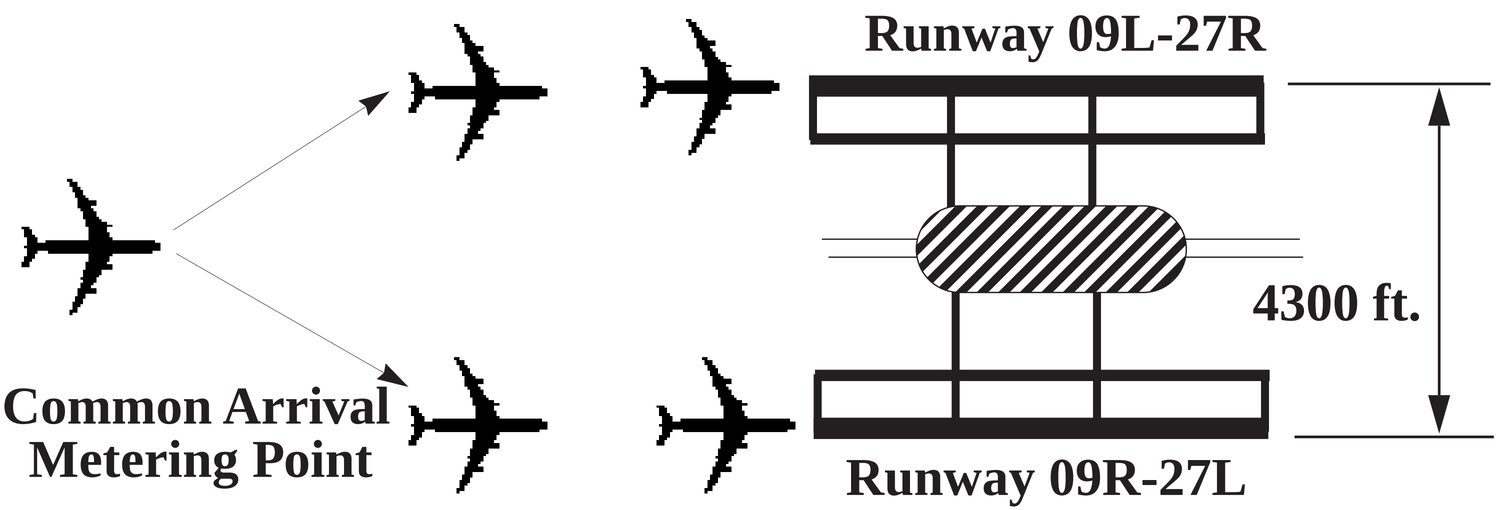
if $s - 1 - \lambda/\mu = 0$ then use

$$\frac{1 - e^{-\mu t(s - 1 - \lambda/\mu)}}{s - 1 - \lambda/\mu} = \mu t$$

# Illustrative Example (Analytic Model)

Assume IFR conditions to a large hub airport with

- Arrival rates to metering point are 45 aircraft/hr.
- Service times dictated by in-trail separations (120 s headways)

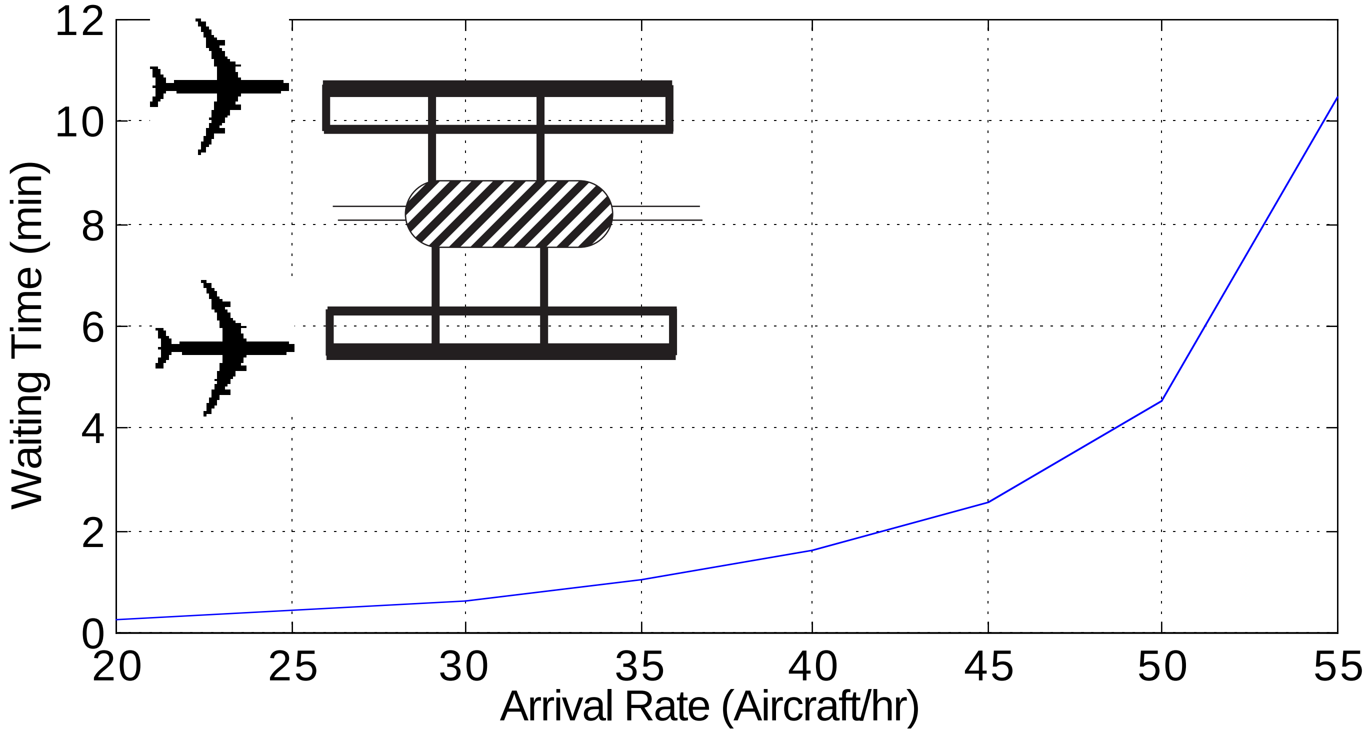


# Some Results of this Simple Model

| Parameter | Numerical Values |
|---|---|
| $\lambda$ | 45 aircraft/hr. to arrival metering point |
| $\mu$ | 30 aircraft per runway per hour |
| $P_o$ | 0.143 |
| $\rho$ | 0.750 |
| $L$ | 3.42 aircraft (includes those in service) |
| $W_q$ | 2.57 minutes per aircraft |
| $W$ | 4.57 minutes per aircraft |

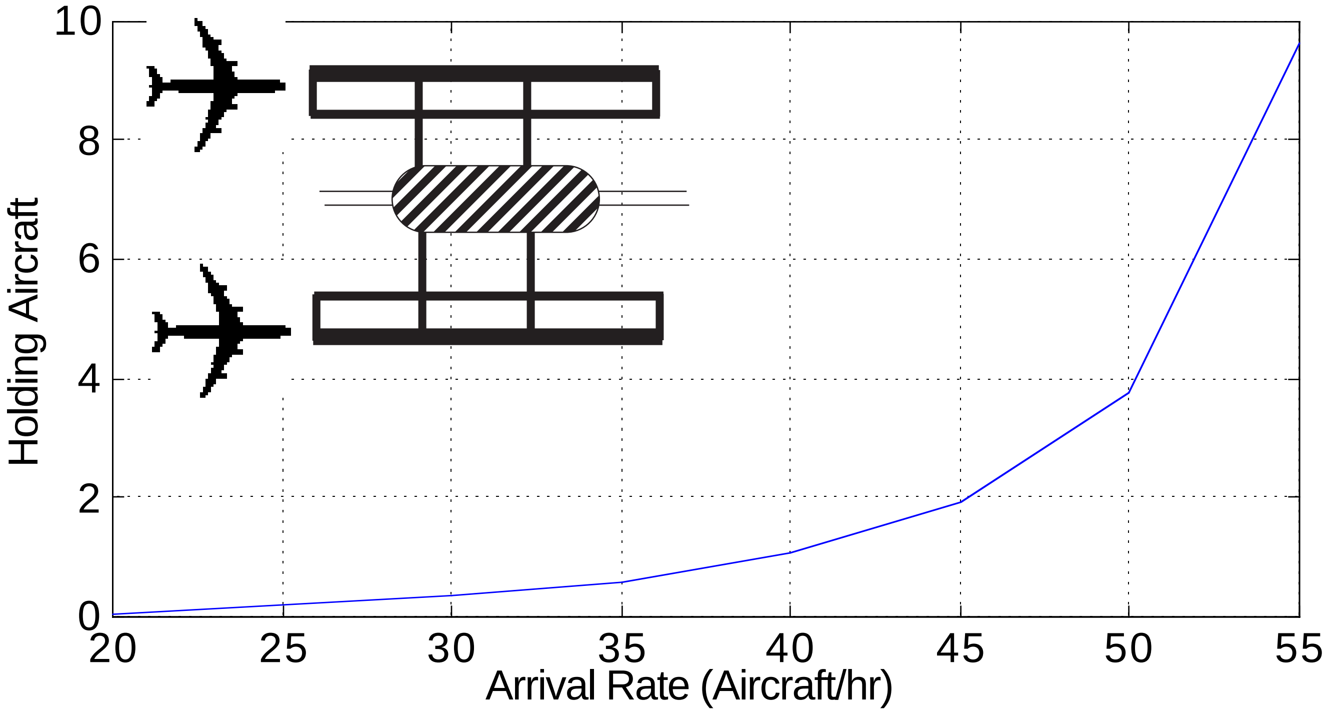
# Sensitivity Analysis

Lets vary the arrival rate (λ) from 20 to 55 per hour and see the effect on the aircraft delay function.



# Sensitivity of $L_q$ with Demand

The following diagram plots the sensitivity of the expected number of aircraft holding vs. the demand function



# Example 2: Level of Service at Airport Terminal Security Checkpoints

The airport shown in the next figures has two security checkpoints for all passengers boarding aircraft. Each security check point has two x-ray machines. A survey reveals that on the average a passenger takes 45 seconds to go through the system (negative exponential distribution service time).

The **arrival rate** is known to be random (this equates to a Poisson distribution) with a mean arrival rate of one passenger every 25 seconds.

In the design year (2010) the demand for services is expected to grow by 60% compared to that today.

# Relevant Operational Questions

a) What is the current utilization of the queueing system (i.e., two x-ray machines)?

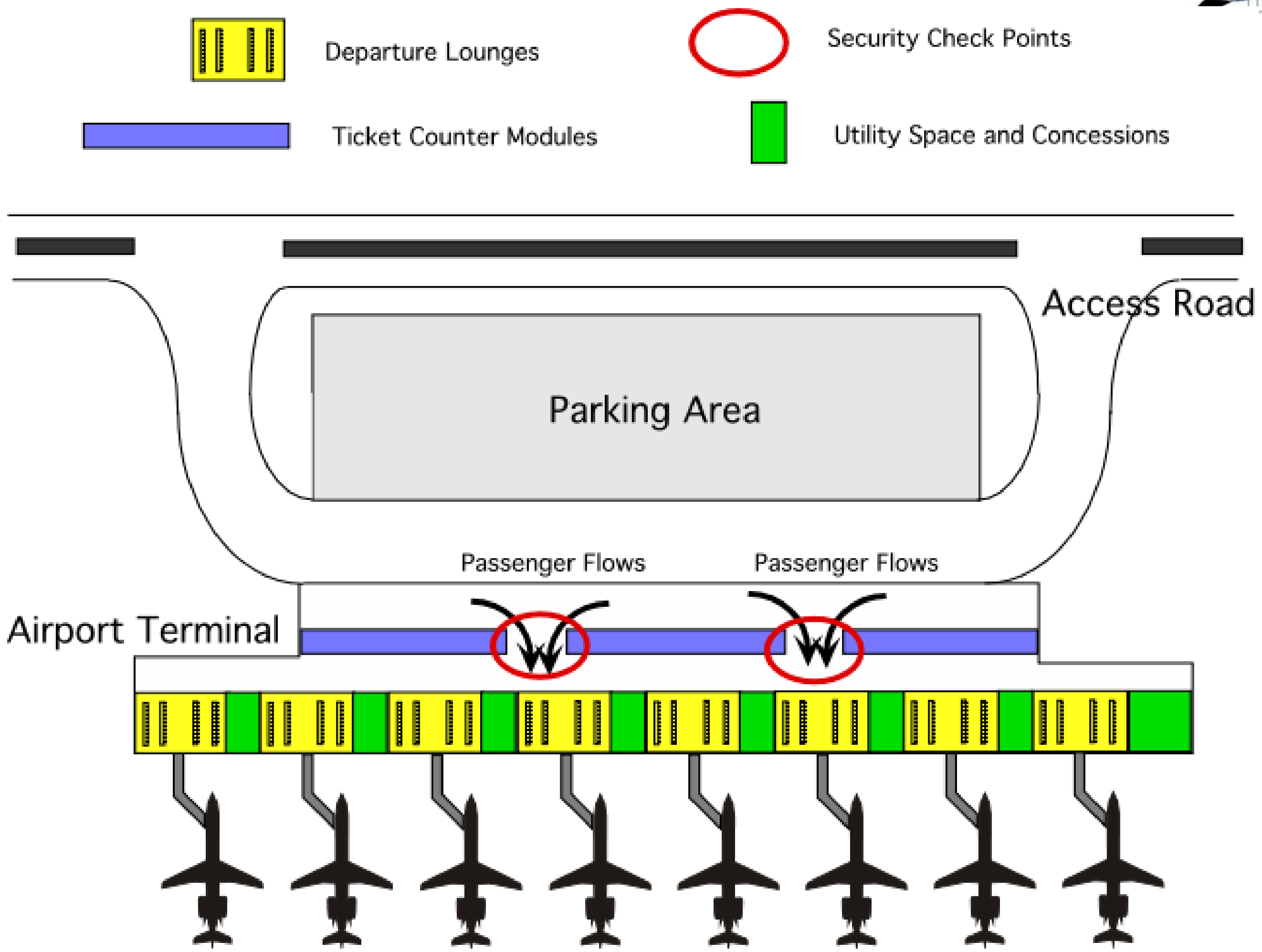
b) What should be the number of x-ray machines for the design year of this terminal (year 2010) if the maximum tolerable **waiting time in the queue** is 2 minutes?

c) What is the expected number of passengers at the checkpoint area on a typical day in the design year (year 2010)?

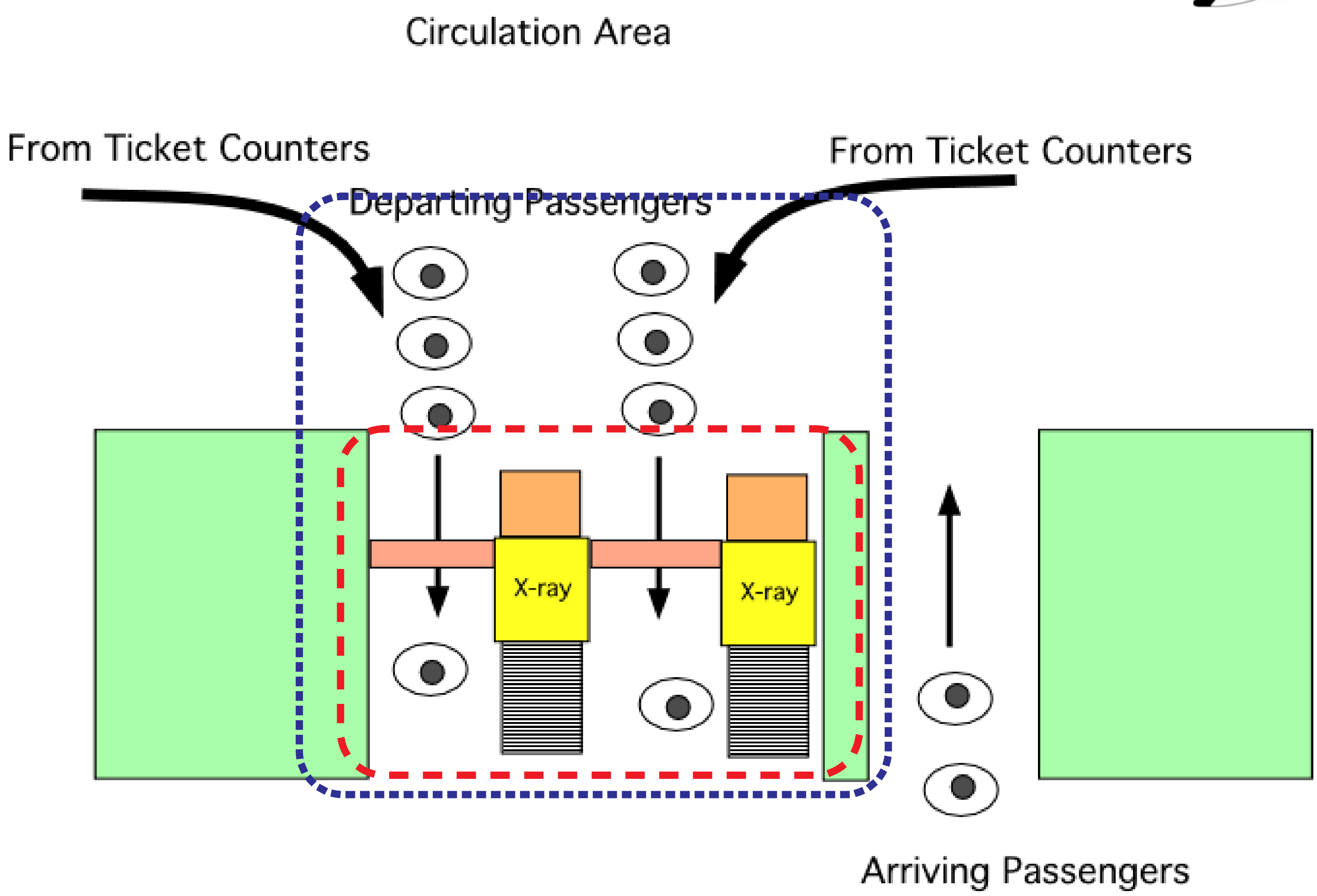
d) What is the new utilization of the future facility?

e) What is the probability that more than 4 passengers wait for service in the design year?

# Airport Terminal Layout



# Security Check Point Layout



# Security Check Point Solutions

a) Utilization of the facility, ρ. Note that this is a multiple server case with infinite source.

$\rho = \lambda / (s\mu) = 140/(2*80) = 0.90$

Other queueing parameters are found using the steady-state equations for a multi-server queueing system with infinite population are:

Idle probability = 0.052632
Expected No. of customers in queue ($L_q$) = 7.6737
Expected No. of customers in system ($L$) = 9.4737
Average Waiting Time in Queue = 192 s
Average Waiting Time in System = 237 s

b) The solution to this part is done by trail and error (unless you have access to design charts used in queueing models. As a first trial lets assume that the number of x-ray machines is 3 (s=3).

Finding Po,

$$P_0 = \sum_{n=0}^{s-1} \frac{(\lambda/\mu)^2}{n!} + \frac{(\lambda/\mu)^s}{s!}\left(\frac{1}{1-(\lambda/s\mu)}\right)$$

Po = .0097 or less than 1% of the time the facility is idle

Find the waiting time in the queue,

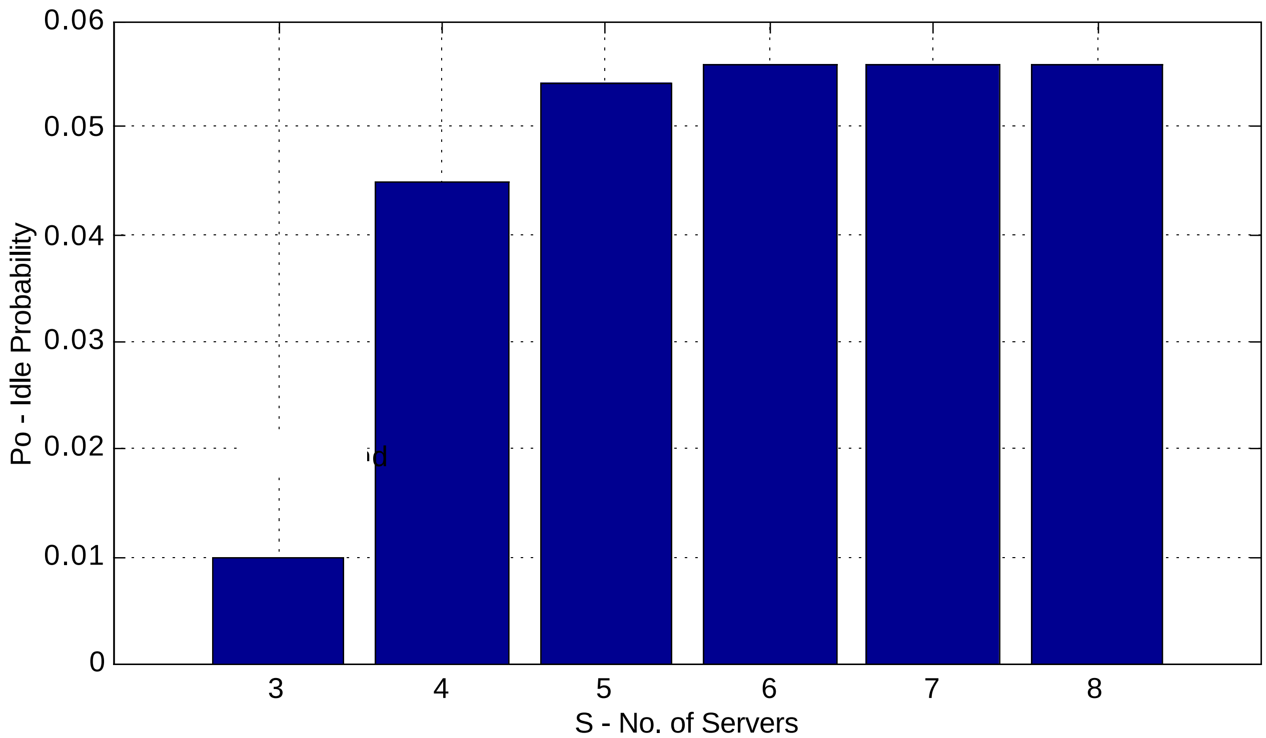
***Wq* = 332** s

Since this waiting time violates the desired two minute maximum it is suggested that we try a higher number of x-ray machines to expedite service (at the expense of cost). The following figure illustrates the sensitivity of $P_o$ and $L_q$ as the number of servers is increased.

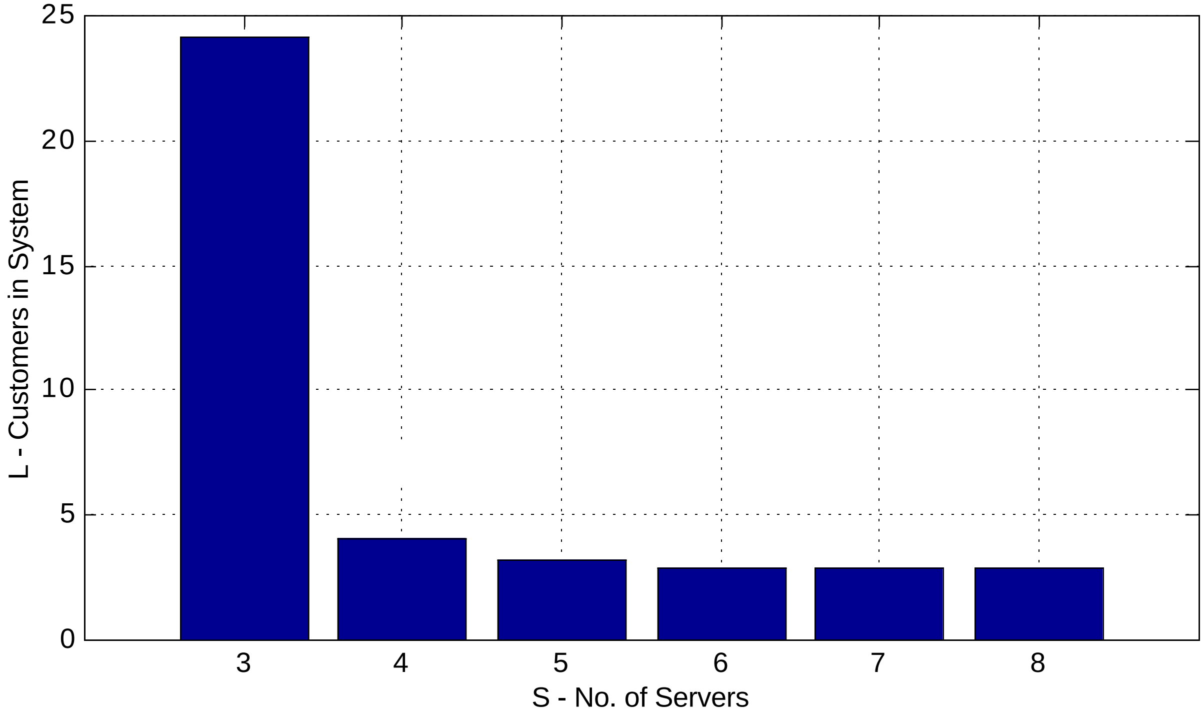
Note that four x-ray machines are needed to provide the desired average waiting time, *Wq*.

# Sensitivity of *Po* with *S*

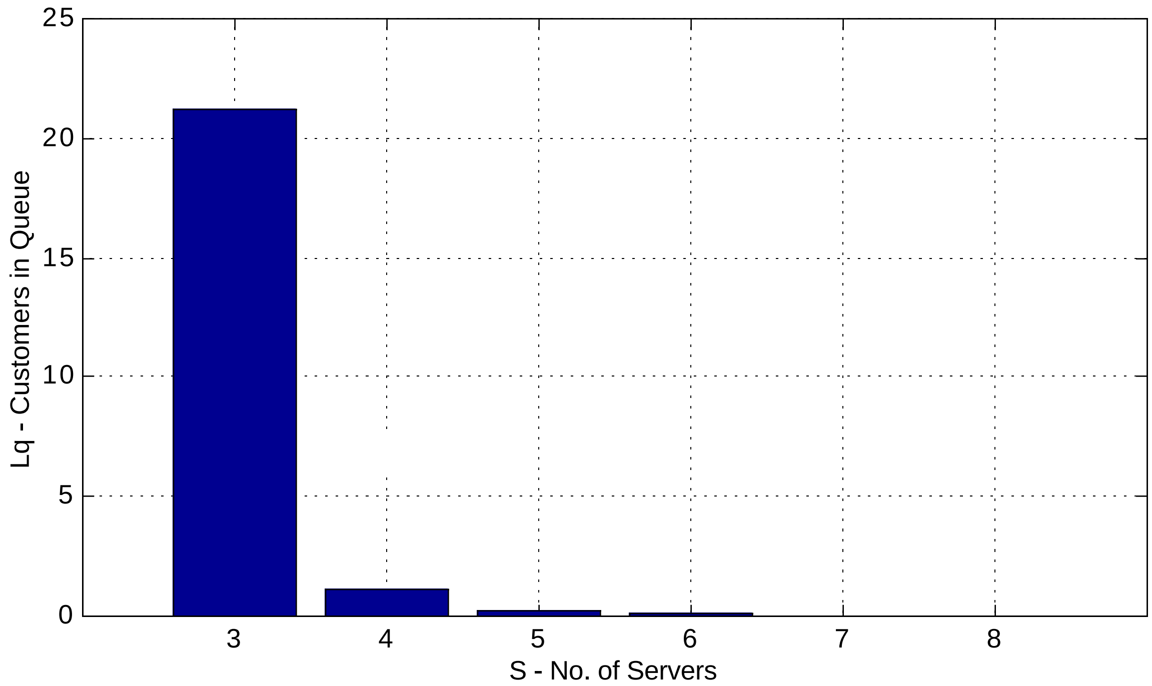
Note the variations in Po as S increases.



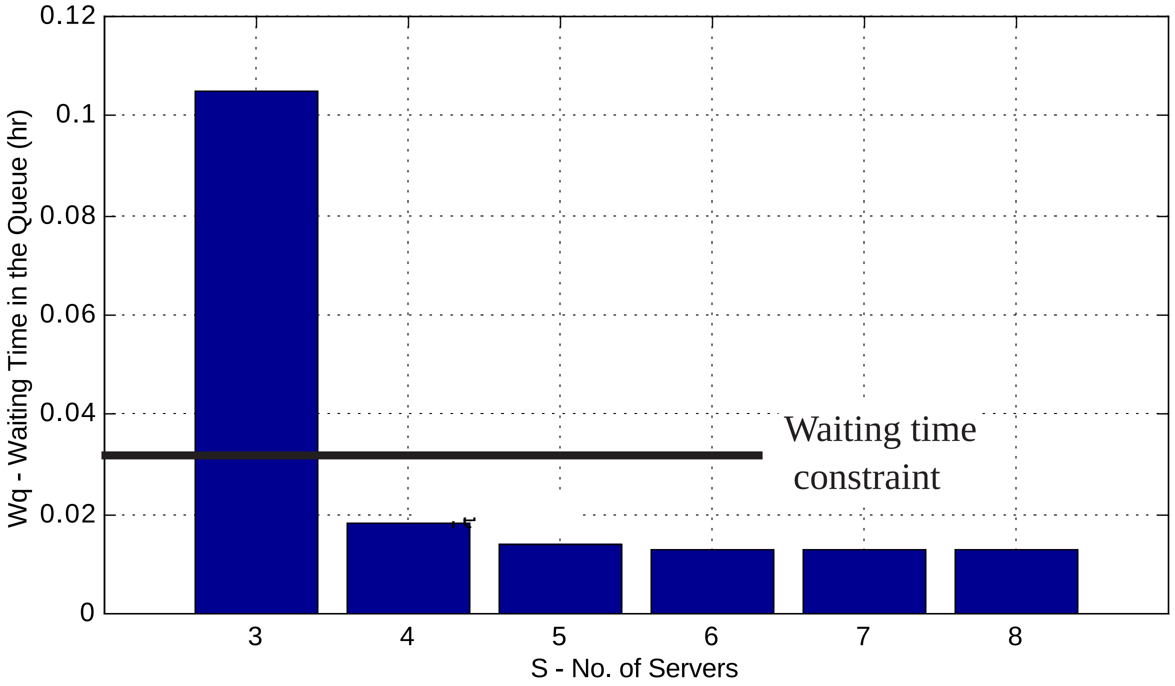
# Sensitivity of *L* with *S*



# Sensitivity of *Lq* with *S*



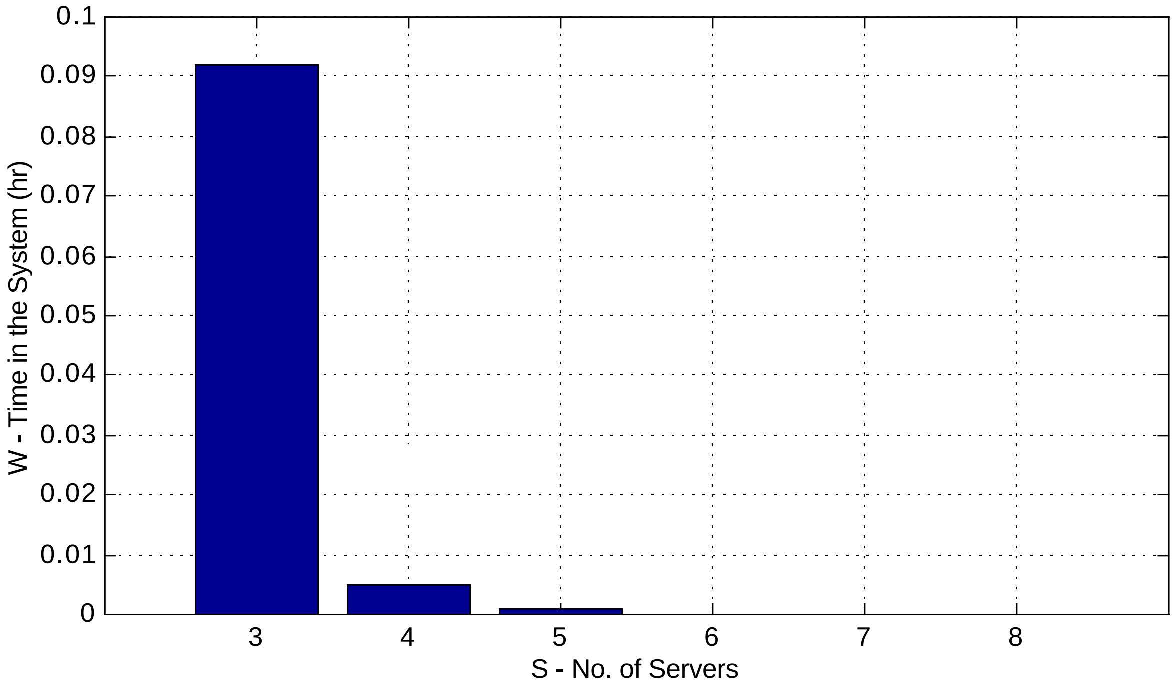
# Sensitivity of *Wq* with *S*




This analysis demonstrates that 4 x-ray machines are needed to satisfy the 2-minute design constraint.

# Sensitivity of *W* with *S*

Note how fast the waiting time function decreases with *S*



# Security Check Point Results

c) The expected number of passengers in the system is (with $S$ = 4),

$$L = \frac{\rho P_0 \left(\frac{\lambda}{\mu}\right)^s}{s!(1-\rho)^2} + \frac{\lambda}{\mu}$$

***L* =** 4.04 passengers in the system on the average design year day.

**d)** The utilization of the improved facility (i.e., four x-ray machines) is

$\rho = \lambda / (s\mu) = 230/ (4*80) = \mathbf{0.72}$

**e)** The probability that more than four passengers wait for service is just the probability that more than eight passengers are in the queueing system, since four are being served and more than four wait.

$$P(n > 8) = 1 - \sum_{n=0}^{8} P_n$$

where,

$$P_n = \frac{(\lambda/\mu)^n}{n!} P_0 \quad \text{if } n \le s$$

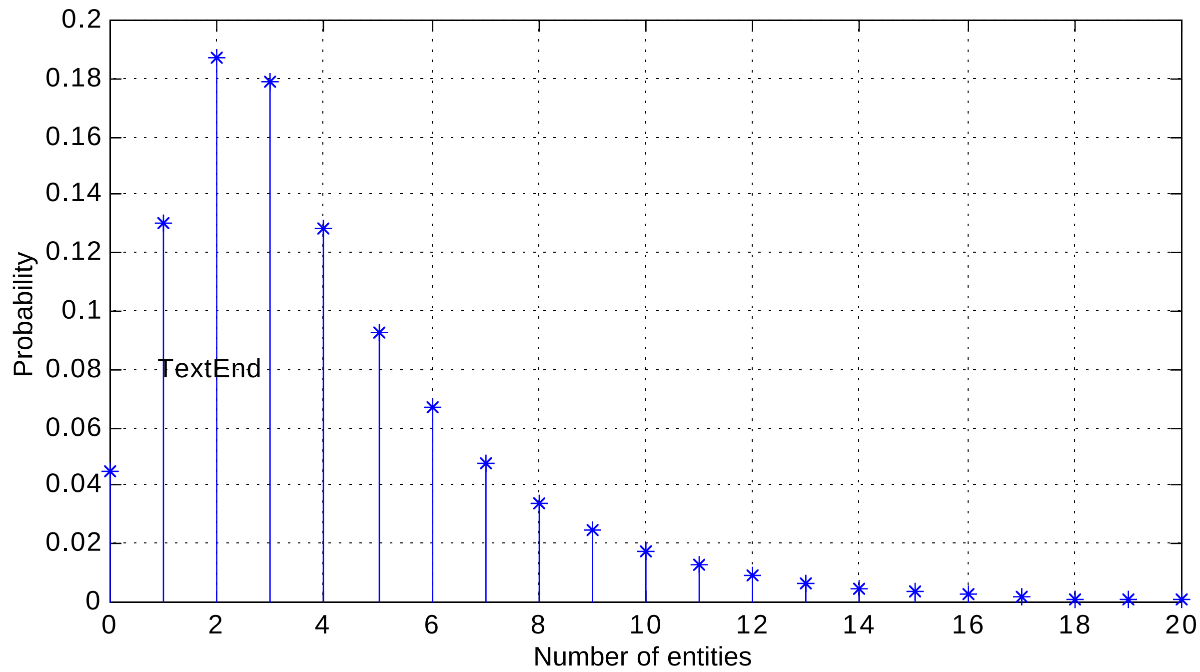
$$P_n = \frac{(\lambda/\mu)^n}{s!s^{n-s}} P_0 \quad \text{if } n > s$$

from where, $P_n > 8$ is 0.0879.

Note that this probability is low and therefore the facility seems properly designed to handle the majority of the expected traffic within the two-minute waiting time constraint.

# PDF of Customers in System (L)

The PDF below illustrates the stochastic process resulting from poisson arrivals and neg. exponential service times



# Conclusions About Analytic Models

Advantages:

- Good traceability of causality between variables
- Good only for first order approximations
- Easy to implement

Disadvantages:

- Too simple to analyze small changes in a complex system
- Cannot model transient behaviors very well
- Large errors are possible because secondary effects are neglected

# Monte Carlo Simulation

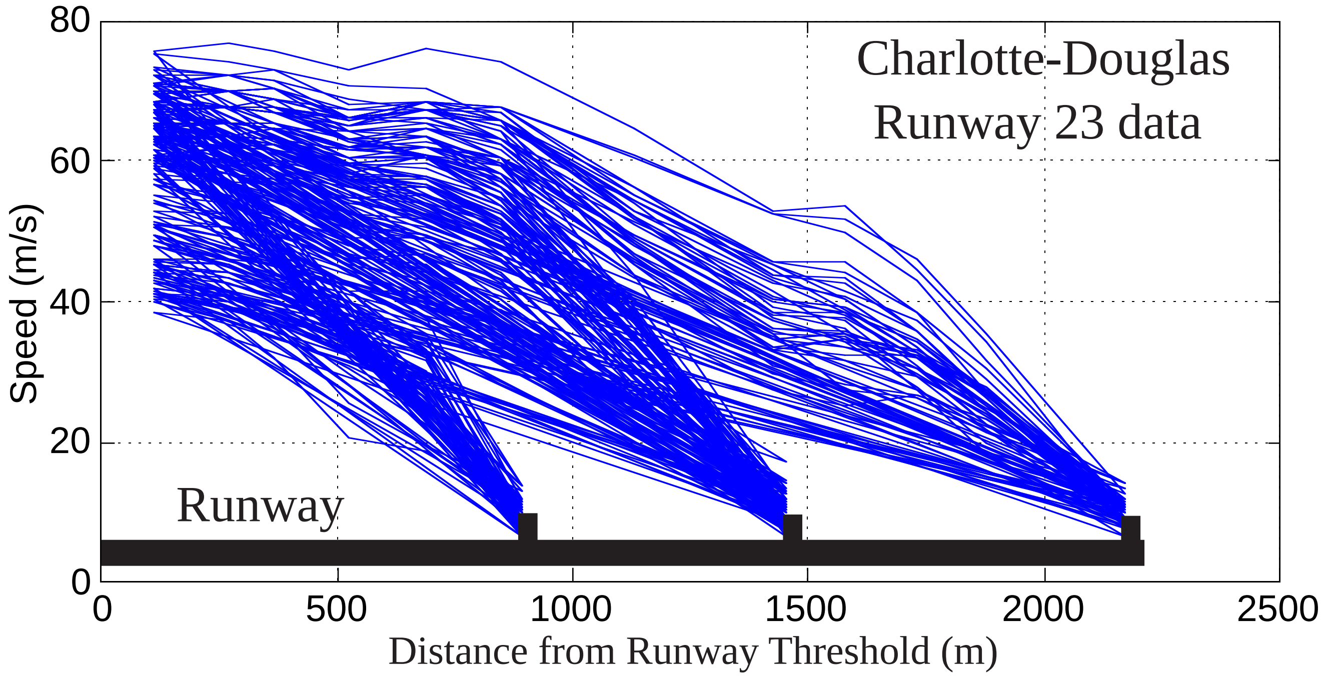
Description of a system with random variables without consideration of the passage of time.

Use of random variates to predict stochastic parameters of a model

In general, time is not a factor in this type of simulation

# Example 3 - Monte Carlo Simulation

Suppose we want to model the stochastic behavior associated with aircraft landings on a runway



# Purpose of the Model

To predict landing distances

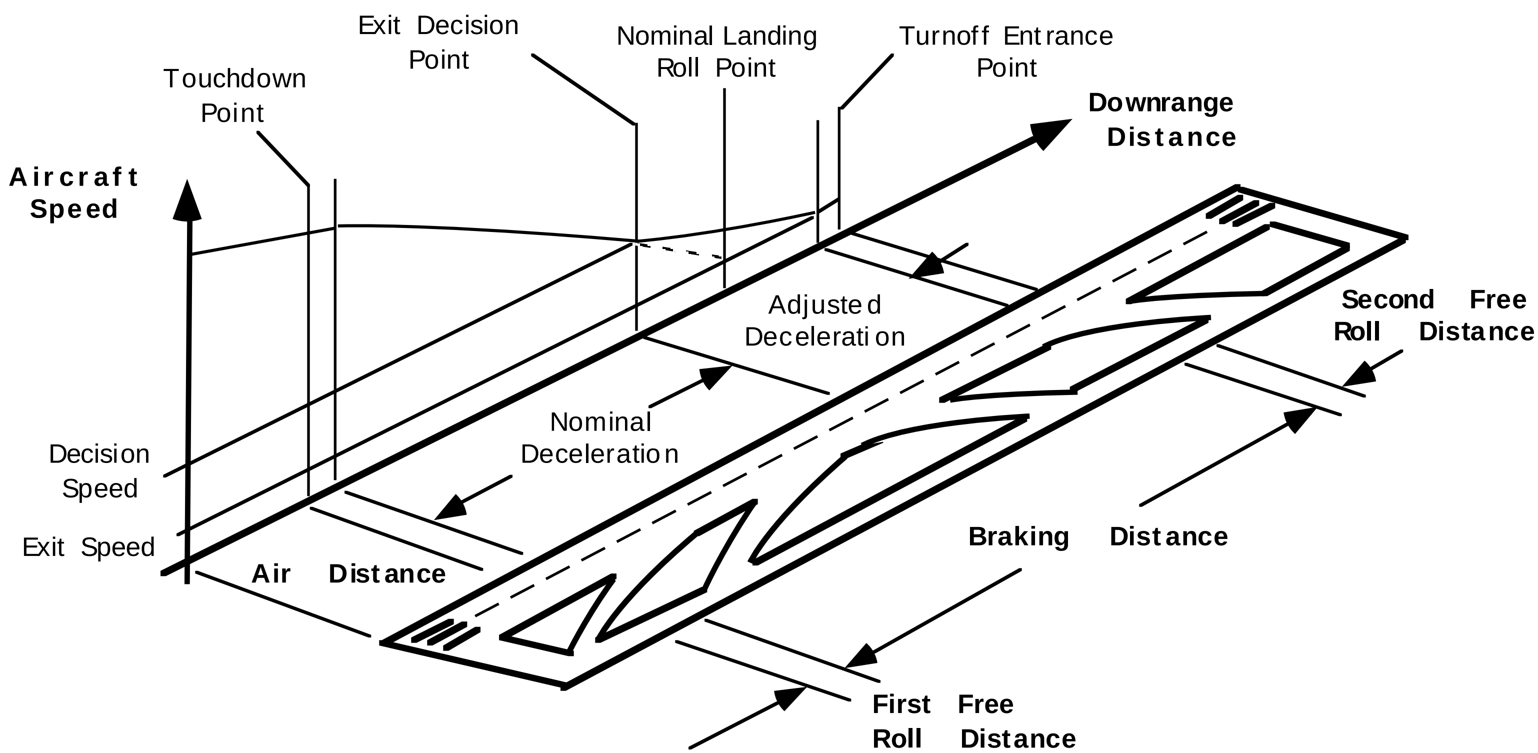
To predict runway occupancy times

To predict runway exit use

To optimize runway exit placement

# Model Abstraction Phase

Assume that five phases (or segments) are considered in the analysis of landing roll operations



# Description of Model Segments

| Phase | Description |
|---|---|
| Flare | From threshold crossing to main landing gear touchdown |
| Free roll | Accounts for pilot reaction times to identify an exit and to activate thrust reversers |
| Braking | Aerodynamic brakes and thrust reversers active |
| Turnoff | Aircraft maneuvers through a suitable runway exit until leaving the imaginary runway edge plane |

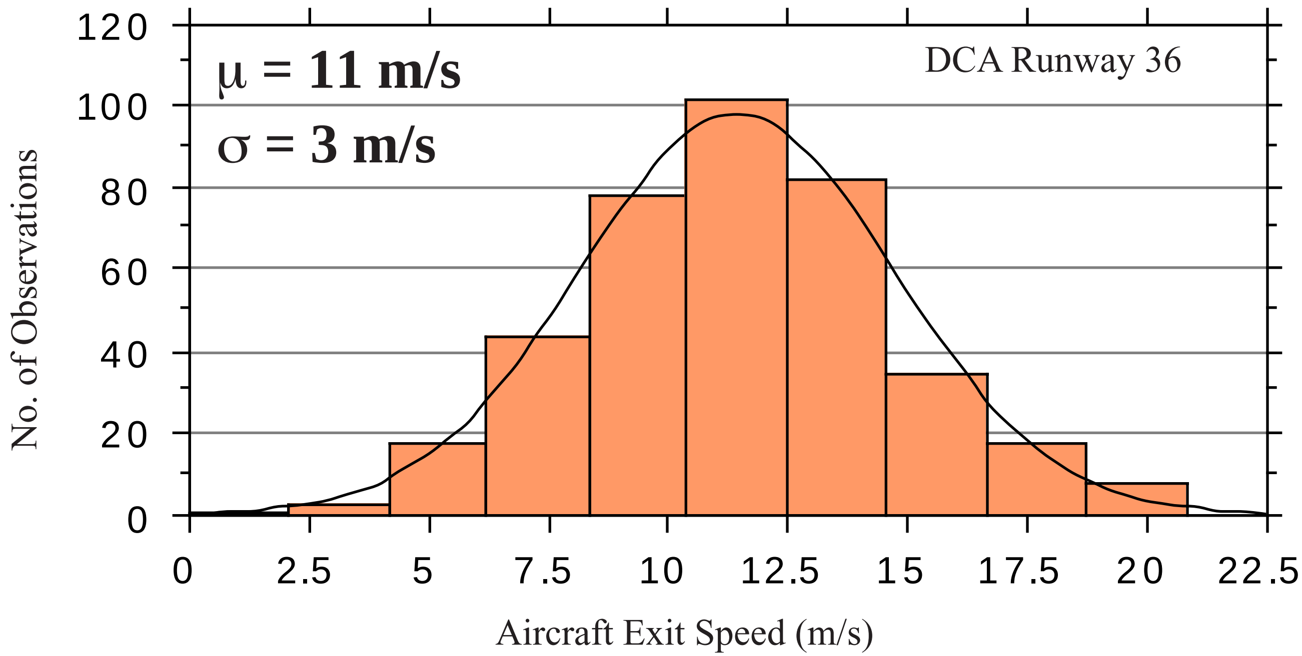
# Key Model Variables

| Phase | Description |
|---|---|
| Flare | Touchdown distance ($d_t$), approach speed ($V_{app}$) |
| Free roll | Reaction time ($t_r$) |
| Braking | Average deceleration ($a_d$) |
| Turnoff | Exit speed ($V_{exit}$), Turnoff time ($t_{turn}$) |

Assume (for simplicity) that each variable is an independent normally distributed random variate with parameters ($\mu$, $\sigma$)

# Example of Normal Distribution R.V.

Data collected supports the conclusion that several of these variables are random variates with reasonable Probability Density Functions (PDFs).



# Monte Carlo Data Generation Step

Boeing 727-200 aircraft data collected at five airports.

| Parameter Description | Distribution Values (μ,σ) |
|---|---|
| Air distance (m) | $d_t$ = (340,120) |
| Approach speed (m/s) | $V_{app}$ = (67,3) |
| Reaction time in free roll (s) | $t_r$ = (1.7,0.5) |
| Deceleration rate (m/s$^2$) | $a_d$ = (-2.8,0.8) |
| Exit speed (m/s) | $V_{exit}$ = (11.0,3.0) |
| Turnoff time (s) | $t_{turn}$ = (9.7, 2.5) |

# Basic Landing Roll Mathematical Model

| Parameter | Mathematical Expression |
|---|---|
| Air distance (m) | $S_{air} = d_t$ |
| Air time (s) | $t_{air} = 2S_{air}/(V_{app} + 0.95V^{\cdot}_{app})$ |
| Free roll distance (m) | $S_{fr1} = t_r(0.95V_{app})$ |
| Braking distance (m) | $S_{brake} = \left[(0.95V_{app})^2 - (V_{exit})^2\right]/-2a_d$ |
| Braking time (s) | $t_{brake} = (0.95V_{app} - V_{exit})/(-a_d)$ |
| Total distance (m) | $S_{total} = S_{air} + S_{fr1} + S_{brake}$ |
| Runway occup. time (s) | $t_{total} = t_{air} + t_r + t_{brake} + t_{turn}$ |

# Matlab Code

```
% Calculation of Aircraft Runway Occupancy Time
% Define aircraft parameters for Boeing 727-200
% Generate five streams of random numbers (normally
                              distributed)
%
% r(i,1) to be used in air distance estimation
% r(i,2) to be used in the approach speed computation
% r(i,3) to be used in the exit speed estimation
% r(i,4) to be used in the deceleration estimation
% r(i,5) to be used in the estimation of exit time

nsim = 1000;        % no. of aircraft replications
                    (arrivals)
```

```
r = randn (5, nsim);% five streams of random
                          numbers

% Initialize variables

Tfr1 = 1.8;
Tfr2 = 1.5;
Vapp_mean = 67;
Vapp_std = 3;
Sair_mean = 450;
Sair_std = 120;
a_mean = 2.8;
a_std = 0.5;
Vexit_mean = 12.0;
Vexit_std = 3.5;
```

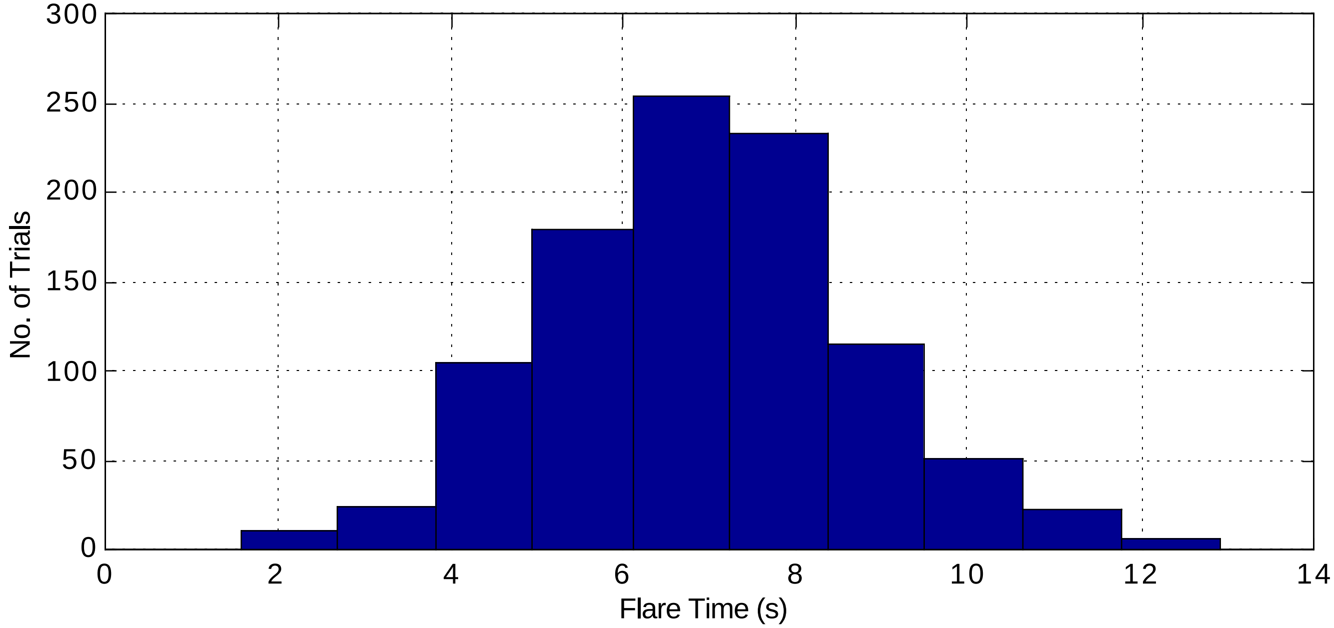
```
Texit_mean = 9.7;
Texit_std = 3.5;

% Estimation of random variates (normal)

i=1:1:nsim;

Vapp = Vapp_mean + Vapp_std * r(1,i);
Sair = Sair_mean + Sair_std * r(2,i);
Tair = 2* Sair./ (Vapp + 0.95*Vapp);

a = a_mean + a_std * r(3,i);
Vexit = Vexit_mean + Vexit_std * r(4,i);
Tbrake = (0.95*Vapp - Vexit)./a;

Texit = Texit_mean + Texit_std * r(5,i);
```

```
ROT = Tair + Tfr1 + Tbrake + Tfr2 + Texit;

bar(i,ROT)
ylabel('Runway occupancy Time (s)')
xlabel('No. of Trial')
grid

pause

hist (Tair)
xlabel('Flare Time (s)')
ylabel('No. of Trials')
grid
pause
```

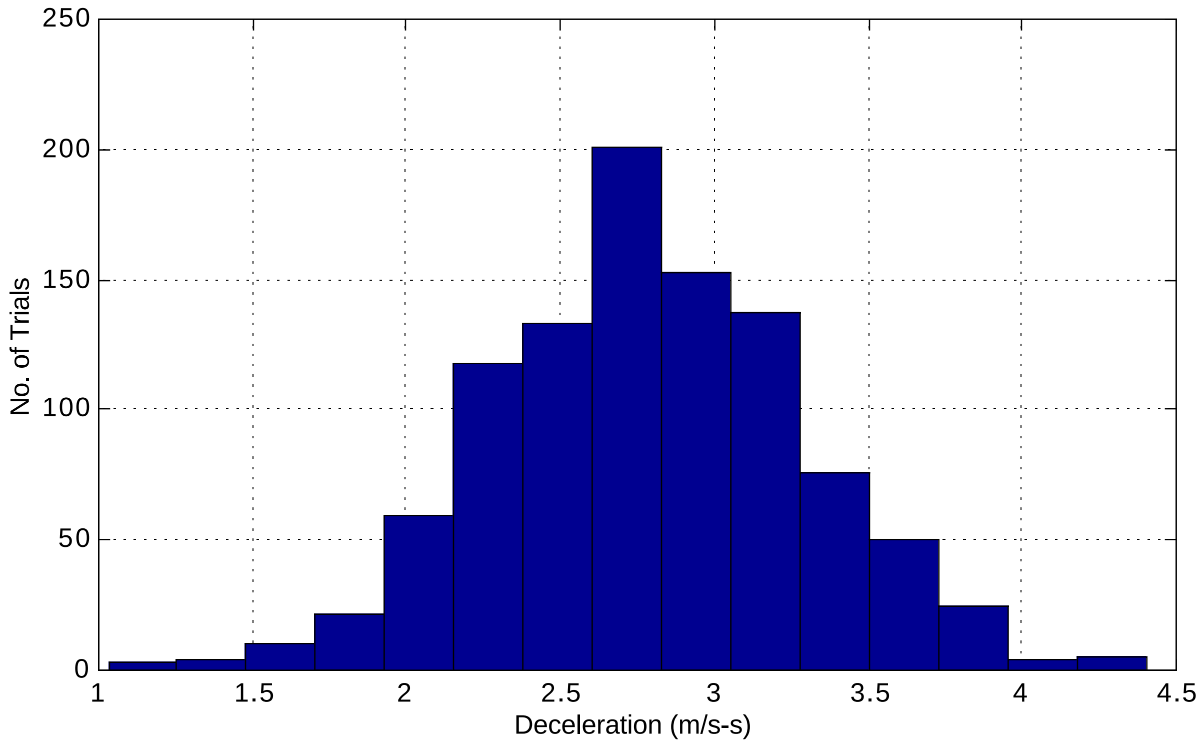
```
hist (a,15)
xlabel('Deceleration (m/s-s)')
ylabel('No. of Trials')
grid

pause

hist (Tbrake)
xlabel('Brake Time (s)')
ylabel('No. of Trials')
grid
pause

hist (Texit)
xlabel('Exit Time (s)')
ylabel('No. of Trials')
```

```
grid

pause

hist (ROT,20)
xlabel('Runway Occupancy Time (s)')
ylabel('No. of Trials'); grid
```

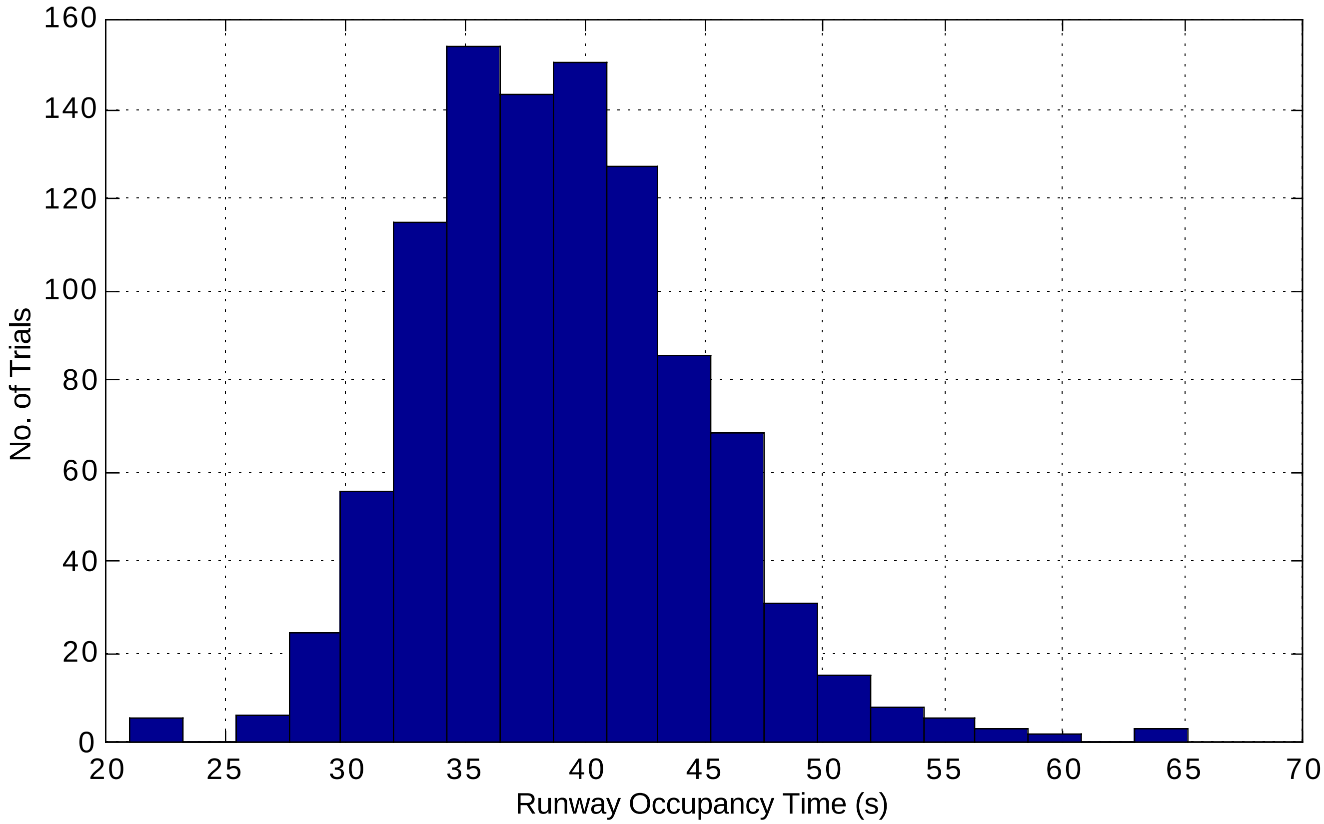
# Sample Results (Flare Time)



# Sample Results (Deceleration)

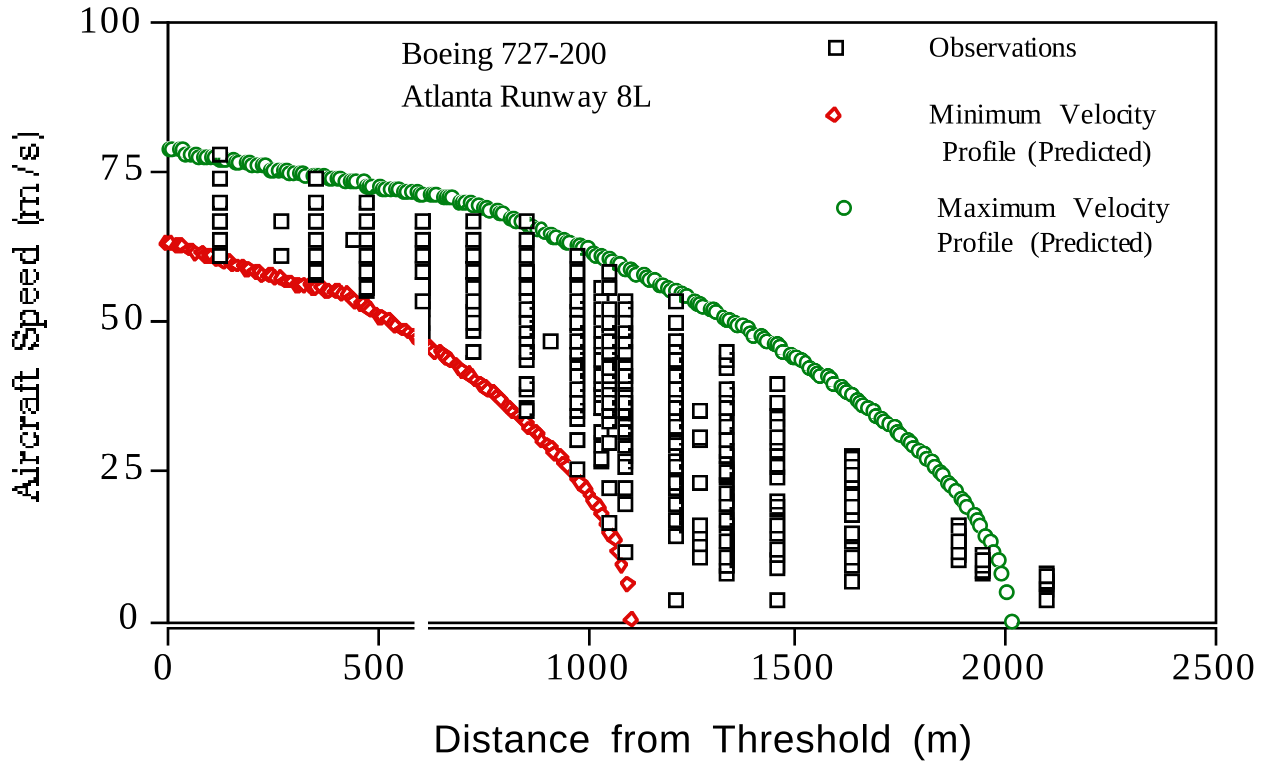


# Results (Runway Occupancy)



# Landing Roll Prediction Model Results

Convolution of observed (assumed) distributions.



# Remarks About Monte Carlo Simulation

## Advantages:

- Good causality between variables if statistical significance is demonstrated
- Good only for first order approximations
- Easy to implement in a personal computer (e.g., spreadsheet, high-level language)
- Results are more realistic than those derived using analytic models

## Disadvantages:

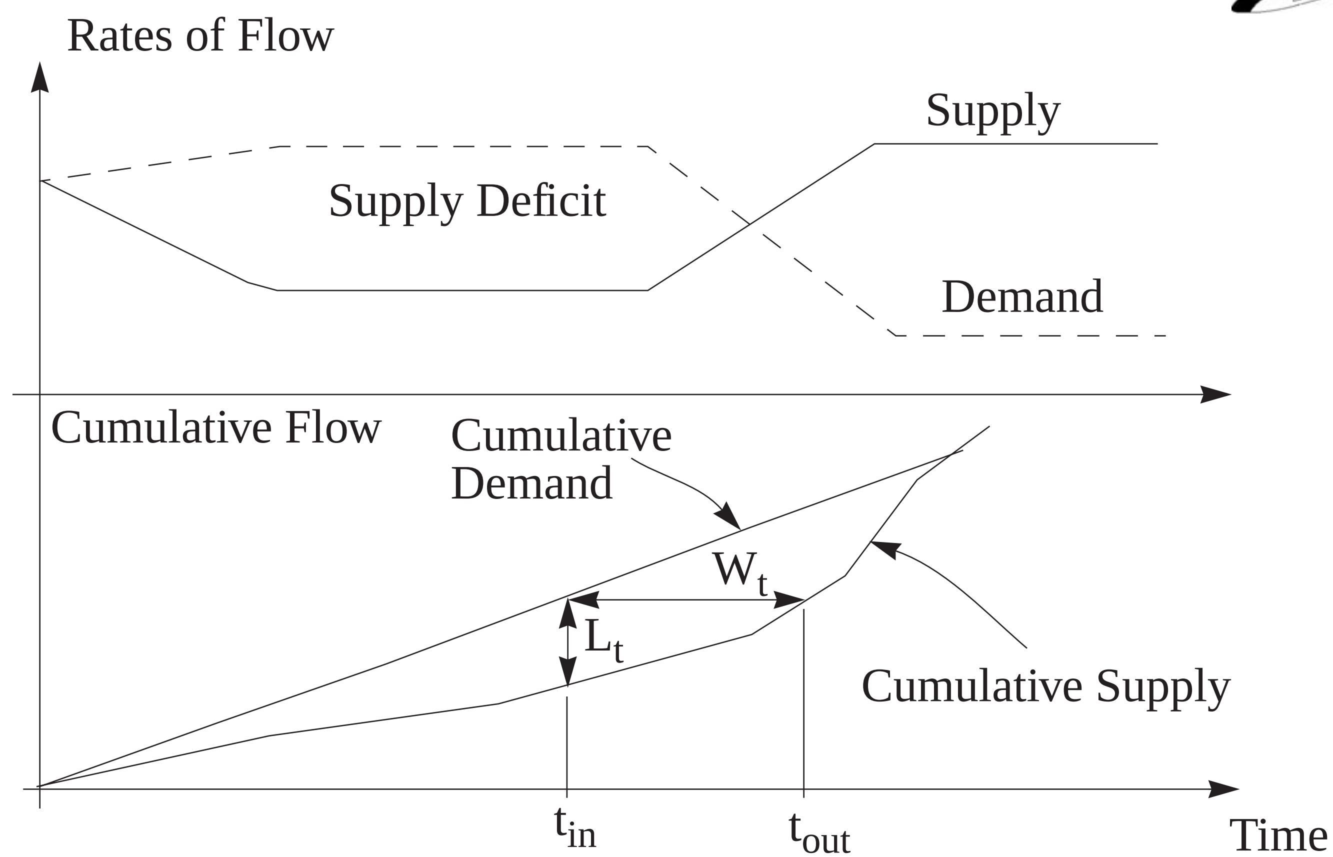
- Time dependencies usually ignored
- Require a PC to get an answer (computational intensive)

# Continuous Simulation

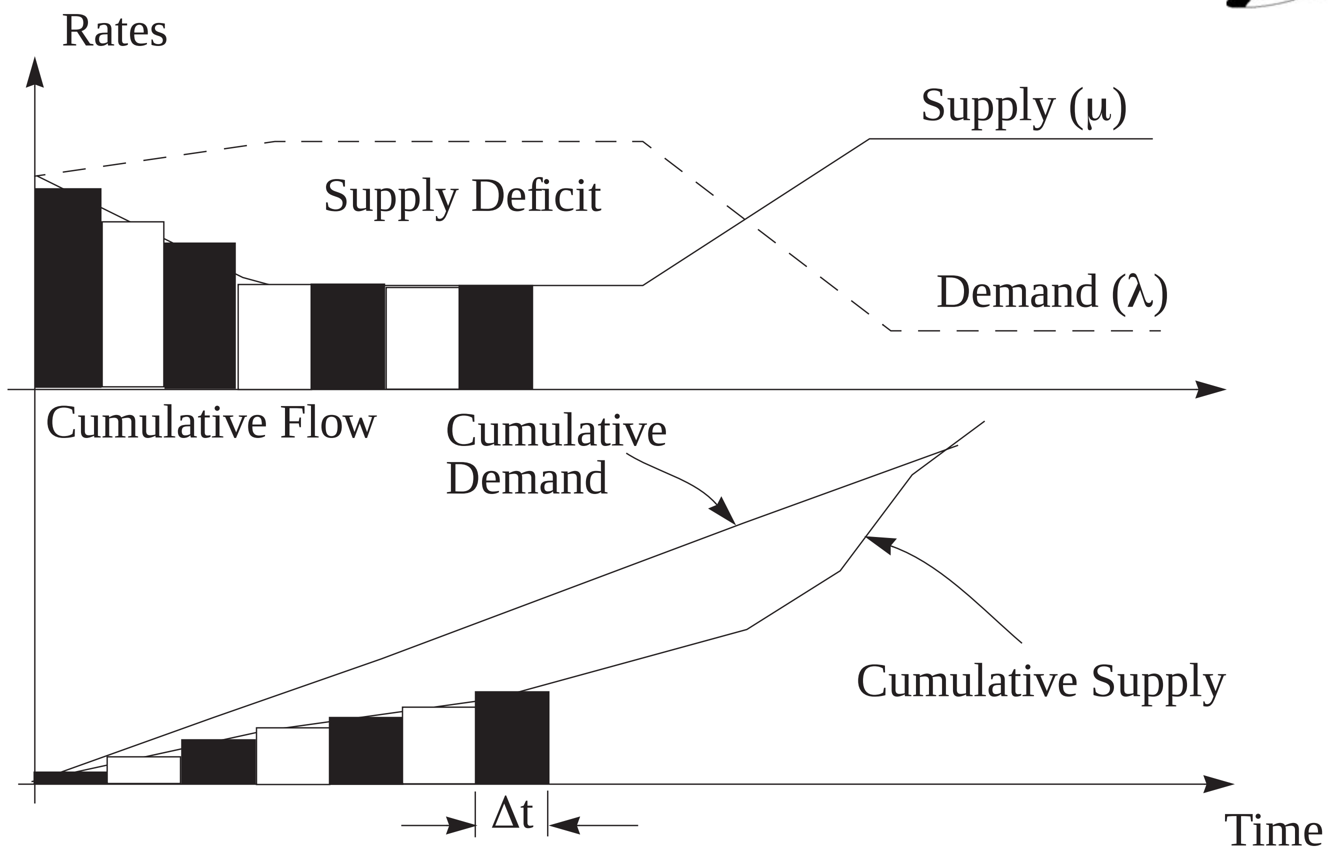
An example of continuous simulation is the modeling of a deterministic queue at an airport terminal where passenger flows are treated as continuous functions of time.

Deterministic Queues are analogous to a continuous flow of entities passing over a facility over time. The figure below depicts graphically a deterministic queue characterized by a region where demand exceeds supply for a given period of time.

# Deterministic Queue



# Estimation of Queueing Parameters



# Deterministic Queue Parameters

- The queue length, $L_t$ , (i.e., state of the system) corresponds to the vertical distance between the cumulative demand and supply curves
- The waiting time, $W_t$, denoted by the horizontal distance between the two cumulative curves in the diagram is the individual waiting time of an entity arriving to the queue at time $t_{in}$
- The total delay is the area under bounded by the cumulative demand and supply curves
- The average delay time is the quotient of the total delay and the number of entities processed

# State of System Definition

Define the state of the system as $L_t$,

$$L_t = \int_0^t (\lambda_t - \mu_t)\,dt$$

$L_t$ is the instantaneous queue length

$\lambda_t$ is the arrival rate function (demand)

$\mu_t$ is the service rate function (supply)

# Differential Equation Representation

Most continuous simulations can be expressed as a set of first order differential equations. The previous state equation for $L_t$ implies:

$$\frac{dL_t}{dt} = (\lambda_t - \mu_t)$$

This equation can be solved numerically (integrating forward with respect to time) if expressed in finite difference form,

$$L_t = L_{t-1} + (\lambda_t - \mu_t)\Delta t$$

# A Word About Integration Algorithms

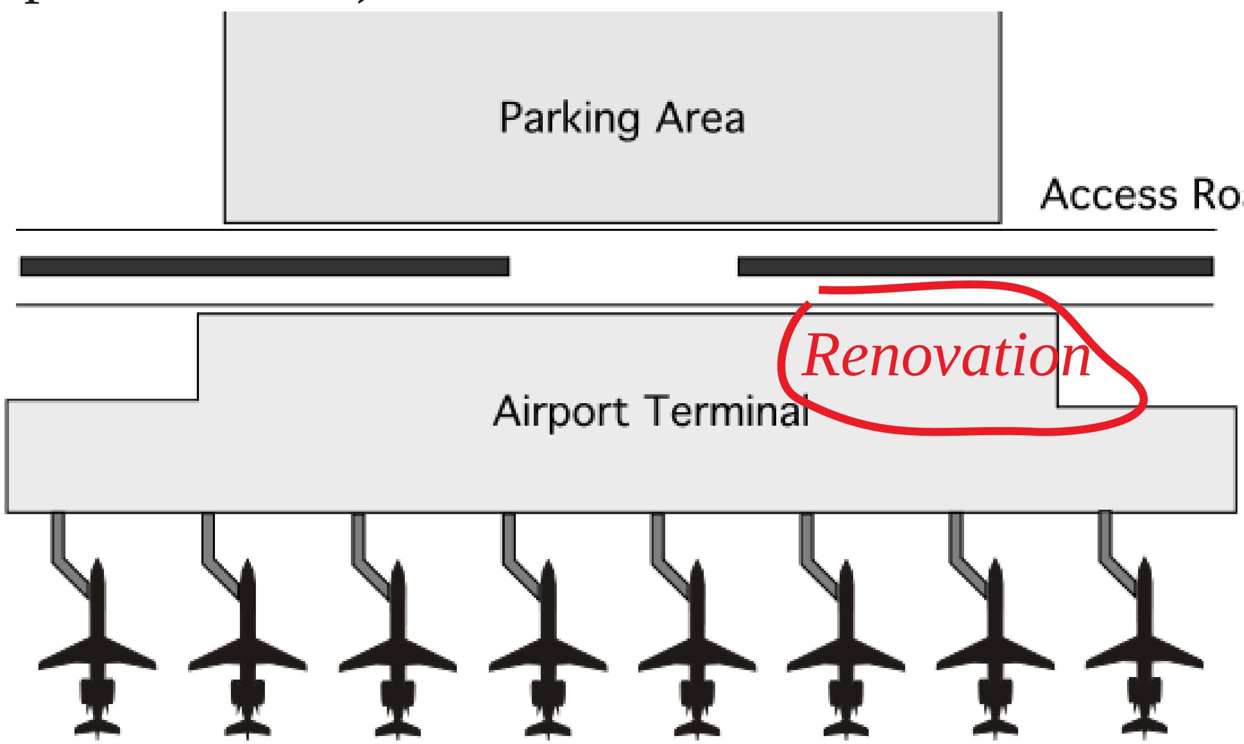
Several techniques can be implemented to solve a set of first order differential equations:

Euler Method - Simplest representation of rate variables (assumes rate variables are constant throughout the integration step size)

Runge- Kutta Methods - Several variations exist of these methods (3rd, 4th, 5th order). Uses a weighted average rate to estimate state variables every integration step. More accurate but more demanding computationally.

# Example 4 - Airport Layout

This example assumes all service areas (ticket counters, security checks, etc.) to be equally spaced inside the airport terminal)



## Mathematical Description of the Problem

$\lambda = 1500$ for $0 < t < 1$

$\lambda = 500$ for $t > 1$

where, $\lambda$ is the arrival function (demand function) and $t$ is the time in hours. Estimate the following parameters:

- The maximum queue length, $L(t)_{\text{max}}$
- The total delay to passengers, $T_d$
- The average length of queue, $L$
- The average waiting time, $W$
- The delay to a passenger arriving 30 minutes hour after the terminal closes for repairs

# Problem Solution (I)

The demand function has been given explicitly in the statement of the problem. The supply function (μ)as stated in the problem is,

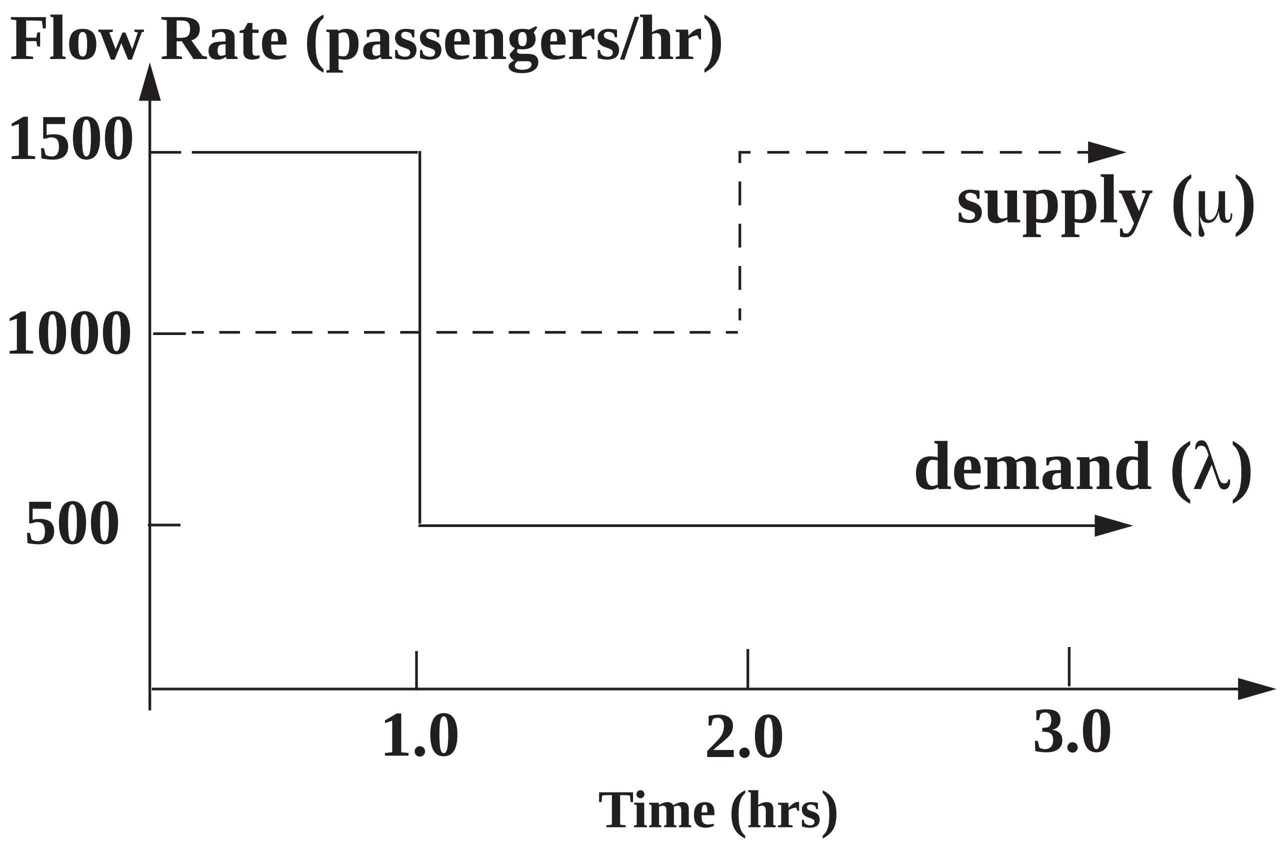
$\mu = 1000$ if $t < 2$

$\mu = 1500$ if $t > 2$

Plotting the demand and supply functions might help understanding the problem

# Problem Solution (II)

Demand and supply functions for the sample problem



# Problem Solution (III)

Sample table simulation using a spreadsheet approach

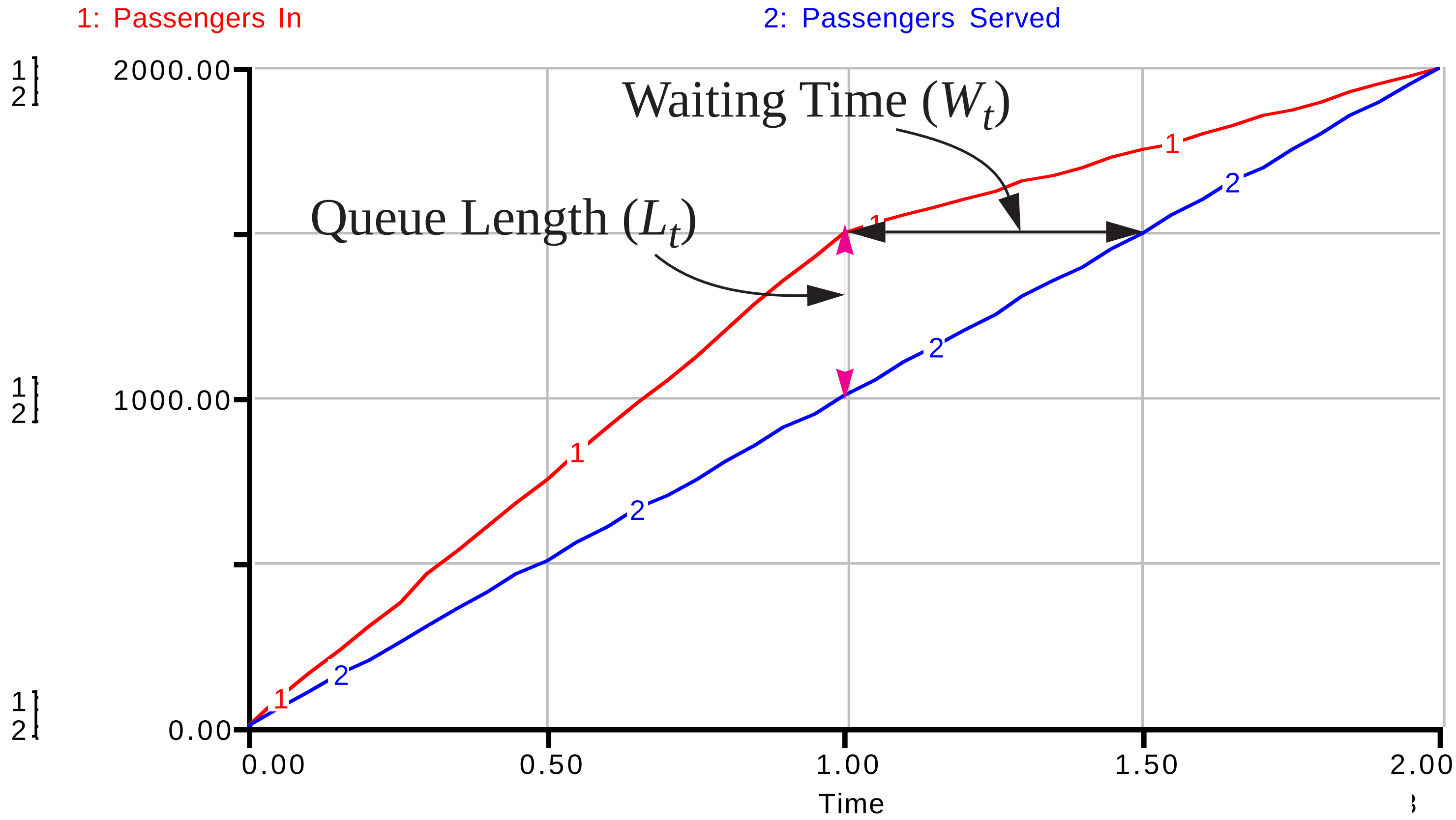
| Simulation Time (hr) | State Variable ($L_t$) | Rate Variable ($\lambda_t$) | Rate Variable ($\mu_t$) | Sum of Rates ($\lambda_t$-$\mu_t$) | (Sum of Rates) $\Delta t$ |
|---|---|---|---|---|---|
| 0 | 0.0 | 1500.0 | 1000.0 | 500.0 | 100.0 |
| 0.2 | 100.0 | 1500.0 | 1000.0 | 500.0 | 100.0 |
| 0.4 | 200.0 | 1500.0 | 1000.0 | 500.0 | 100.0 |
| 0.6 | 300.0 | 1500.0 | 1000.0 | 500.0 | 100.0 |
| 0.8 | 400.0 | 1500.0 | 1000.0 | 500.0 | 100.0 |
| 1.0 | 500.0 | 500.0 | 1000.0 | -500.0 | -100.0 |

| **Simulation Time (hr)** | **State Variable ($L_t$)** | **Rate Variable ($\lambda_t$)** | **Rate Variable ($\mu_t$)** | **Sum of Rates ($\lambda_t$-$\mu_t$)** | **(Sum of Rates) $\Delta t$** |
|---|---|---|---|---|---|
| 1.2 | 400.0 | 500.0 | 1000.0 | -500.0 | -100.0 |
| 1.4 | 300.0 | 500.0 | 1000.0 | -500.0 | -100.0 |

This procedure uses Euler's Method to estimate state variables (i.e., rates $\lambda_t$ and $\mu_t$ are assumed constant throughout every numerical integration interval).

# Problem Solution (IV)

Cumulative flow plots can help visualize the problem



# Problem Solution (V)

The average queue length ($L$) during the period of interest, we evaluate the total area under the cumulative curves (to find total delay)

$T_d$ = 2 [(1/2)(1500-1000)] = 500 passengers-hour

a) The maximum number of passengers in the queue, $L(t)_{\max}$,

$L(t)_{max}$ = 1500 - 1000 = 500 passengers at time $t$=1.0 hours

Find the average delay to a passenger ($W$)

# Problem Solution (VI)

$$W = \frac{T_d}{N_d} = 15 \text{ minutes}$$

where, $T_d$ is the total delay and $N_d$ is the number of passengers that where delayed during the queueing incident.

$$L = \frac{T_d}{t_q} = 250 \text{ passengers}$$

where, $T_d$ is the total delay and $t_d$ is the time that the queue lasts.

# Problem Solution (VII)

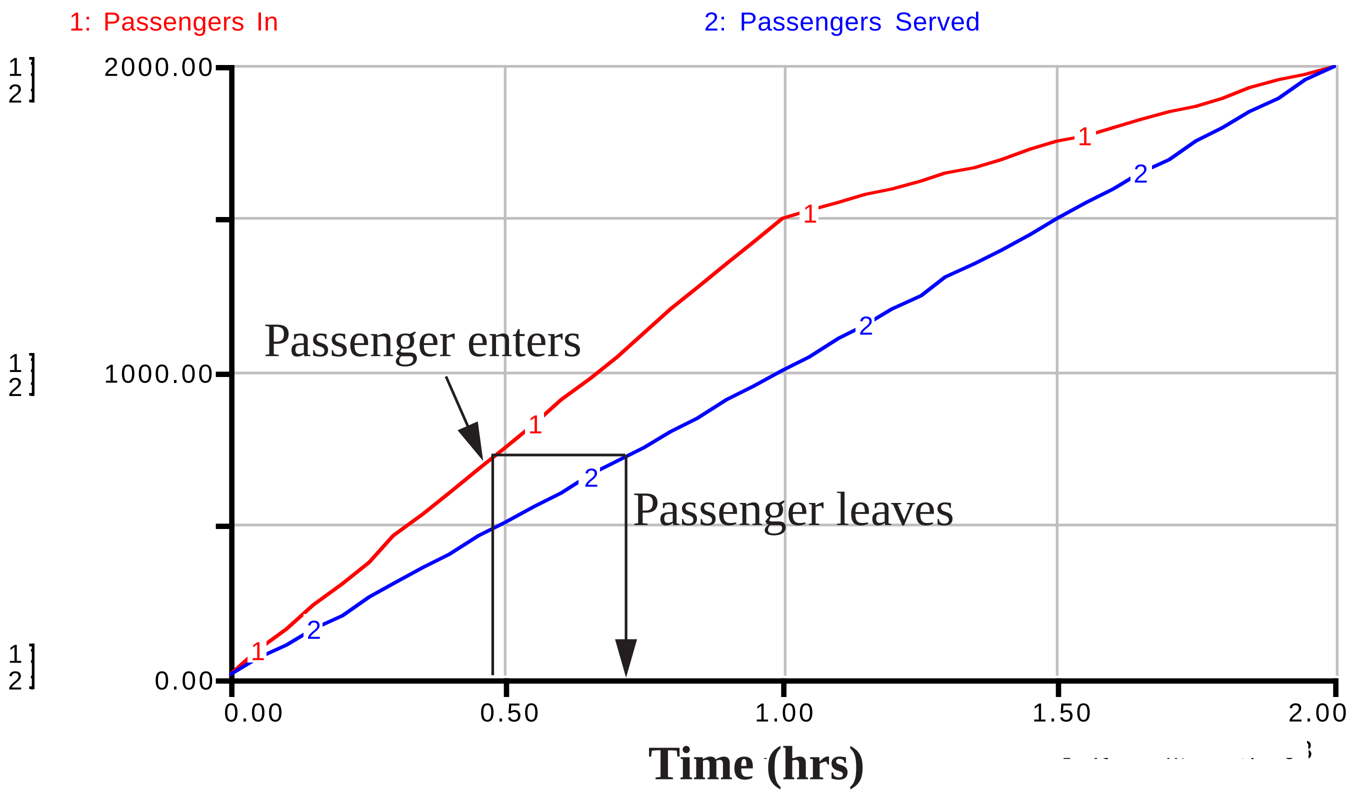
Now we can find the delay for a passenger entering the terminal 30 minutes after the partial terminal closure occurs. Note that at t = 0.5 hours 750 passengers have entered the terminal before the passenger in question. Thus we need to find the time when the supply function, μ(t), achieves a value of 750 so that the passenger "gets serviced". This occurs at,

$$\mu(t + \Delta t) = \lambda(t) = 750$$

therefore Δt is just 15 minutes (the passenger actually leaves the terminal at a time $t$+Δt equal to 0.75 hours). This can be shown in the diagram on the next page.

# Problem Solution (VIII)

Demand and supply functions for example problem



# Handling Complex Time-Varying Behaviors

The methodology described in previous pages can be extended to understand complex airport time-varying behaviors.

Examination of the basic state equation,

$$L_t = L_{t-1} + (\lambda_t - \mu_t)\Delta t$$

reveals that as long as the arrival and service flow rates (i.e., $\lambda_t$ and $\mu_t$ are known functions of time - regardless their mathematical complexity - the process of finding the state, $L_t$, is simple using numerical integration.

# Example 5 - Chicago O'Hare Deterministic Simulation

The following example implements the deterministic queueing equations for a single airport (Chicago O'Hare Intl. Airport - ORD).

The data sets needed to run thus example are extracted from CODAS - the Consolidated Operations and Delay Analysis System database maintained by the Federal Aviation Administration (FAA).

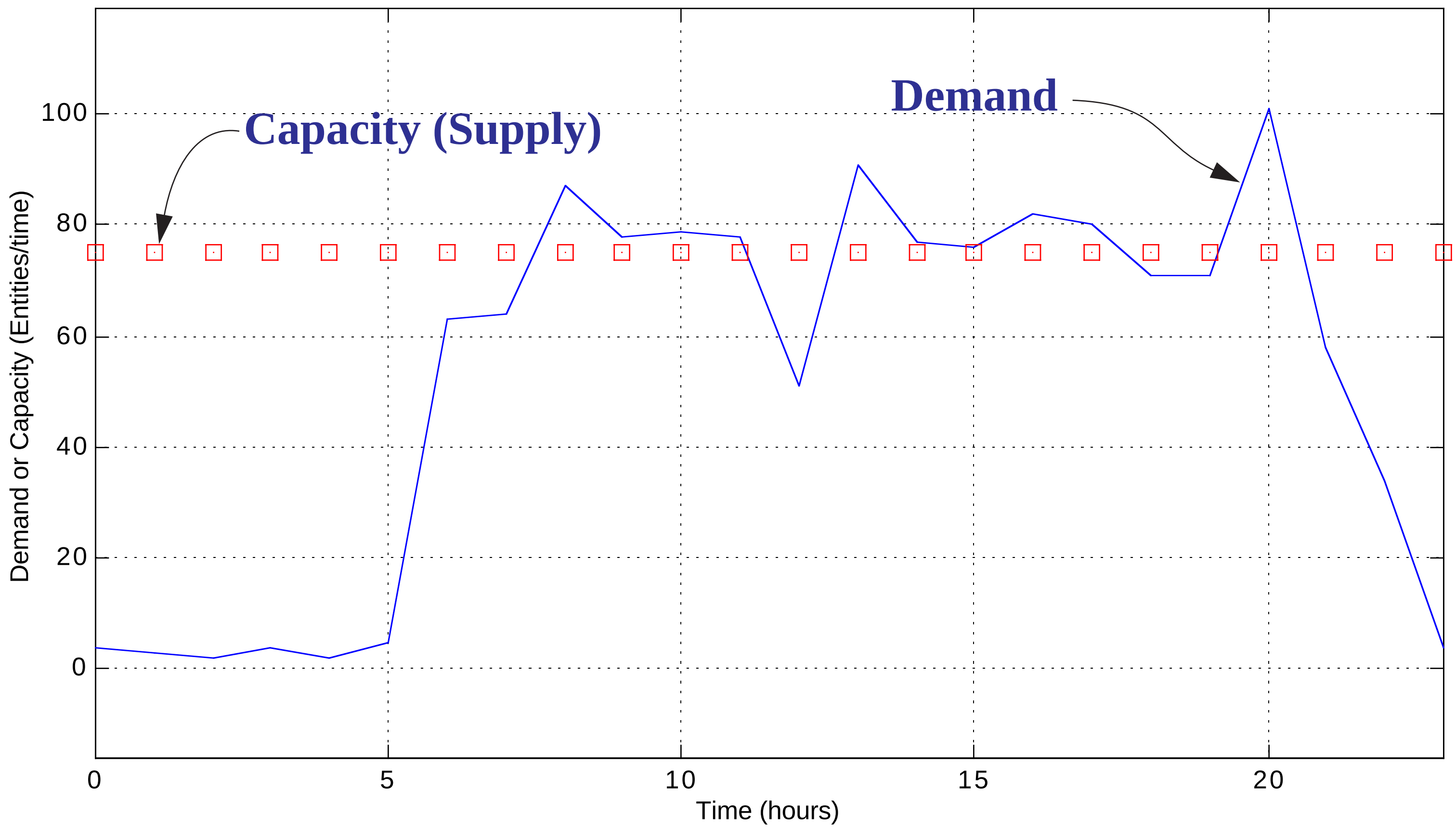
Assumptions:

a) The hourly acceptance rate of the airport is constant (this means $\mu_t$ is constant over time) at 75 operations per hour.

b) The arrival rate function $\lambda_t$ is variable over time and extracted from a real schedule at ORD.

The following figure illustrates graphically the data for this problem.

# ORD Supply and Demand Functions



# Matlab Code Implementation

```
% Deterministic queueing simulation of an airport facility
% Treats demand and capacity as continuous functions
% (time dependent)
%
% Integrates the demand and supply flow rates to get
% cumulative supply and demand curves
%
% Basic state equations =>


% p(1) = L(t) = L(t-1) + [demand(t) - supply(t)] (dt)        % queue length
% p(2) = A(t) = A(t-1) + [L(t)] (dt)                         % area under L(t) curve
%
%  Demand / Supply
```

```
% Programmer: Toni Trani
% Date: October 7, 1999
%
% Calls: fqueue2.m
%
%  fqueue2.m computes the integrals of the demand and supply rates
%  of change
%
% Define global variables
global demand capacity time

% Enter demand function as an array of values over time
% Extracted from CODAS

load ohare_schedule
```

```
[n,m] = size (ohare_schedule);

i=1:n;
time = ohare_schedule(:,1);
demand = ohare_schedule(:,2)
capacity = 75;      % airport arrival capacity per hour

% Compute min and maximum values for proper scaling in plots
mintime = min(time);
maxtime = max(time);
maxd = max(demand);
maxc = max(capacity);
minxd= min(demand);
minc = min(capacity);
```

```
scale                                   = round(.2
*(maxc+maxd)/2)

minplot = min(minc,mind) - scale;

maxplot                                 = max(maxc,maxd) + scale;


po = [0 0];                             % intial number of aircraft

to = mintime;

tf = maxtime;

tspan = [to tf];


% where:

% to is the initial time to solve this equation

% tf is the final time

% tspan is the time span to solve the simulation


[t,p] = ode23('fqueue_2',tspan,po);
```

```
% Compute statistics

Ltmax                = max(p(:,1));
tdelay               = max(p(:,2));
a_demand             = mean(demand);
a_capacity           = mean(capacity);

 clc
 disp([blanks(5),'Deterministic Queueing Model '])
 disp(' ')
 disp(' ')
 disp([blanks(5),'  Average arrival rate (entities/time) = ', num2str(a_demand)])
 disp([blanks(5),'  Average capacity (entities/time) = ', num2str(a_capacity)])
 disp([blanks(5),'  Simulation Period (time units) = ', num2str(maxtime)])
```

```
 disp(' ')

 disp(' ')
 disp([blanks(5),'  Total delay (entities-time) = ', num2str(tdelay)])
 disp([blanks(5),'  Max queue length (entities) = ', num2str(Ltmax)])
 disp(' ')

pause

% Plot the demand and supply functions

subplot(2,1,1)
plot(time,demand,'b',time,capacity,'g')
xlabel('Time (hours)')
ylabel('Demand or Capacity (Entities/time)')
axis([mintime maxtime minplot maxplot])
```

```
grid

subplot(2,1,2)
plot(t,p(:,1),'b')
xlabel('Time')
ylabel('Entities in Queue')
grid

pause

% Plot the results of the numerical integration procedure

subplot(2,1,1)
plot(t,p(:,1),'b')
xlabel('Time')
```

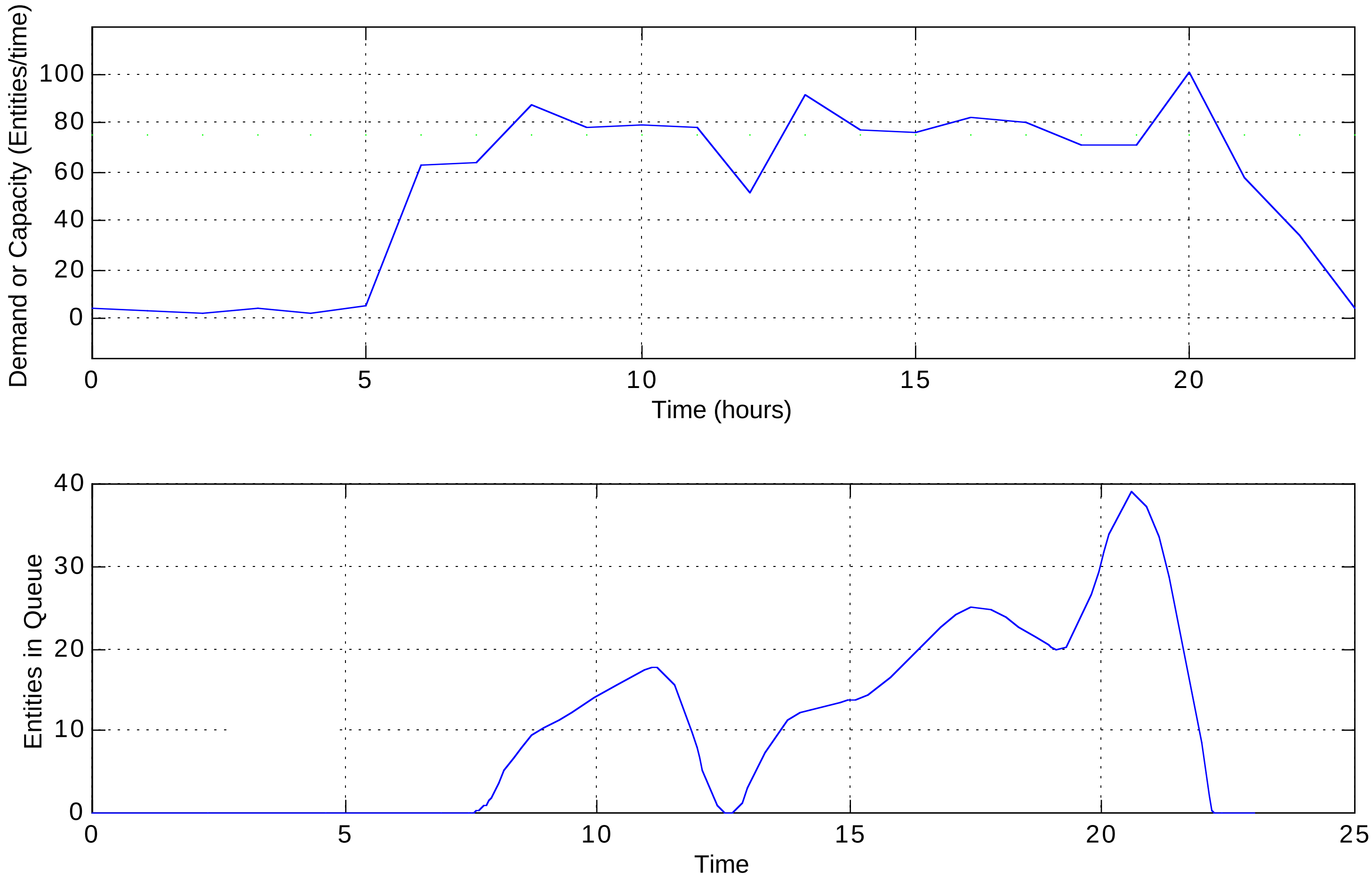
```
ylabel('Entities in Queue')
grid

subplot(2,1,2)
plot(t,p(:,2),'k')
xlabel('Time')
ylabel('Total Delay (Entities-time)')
grid
```

# Function fqueue_2.m

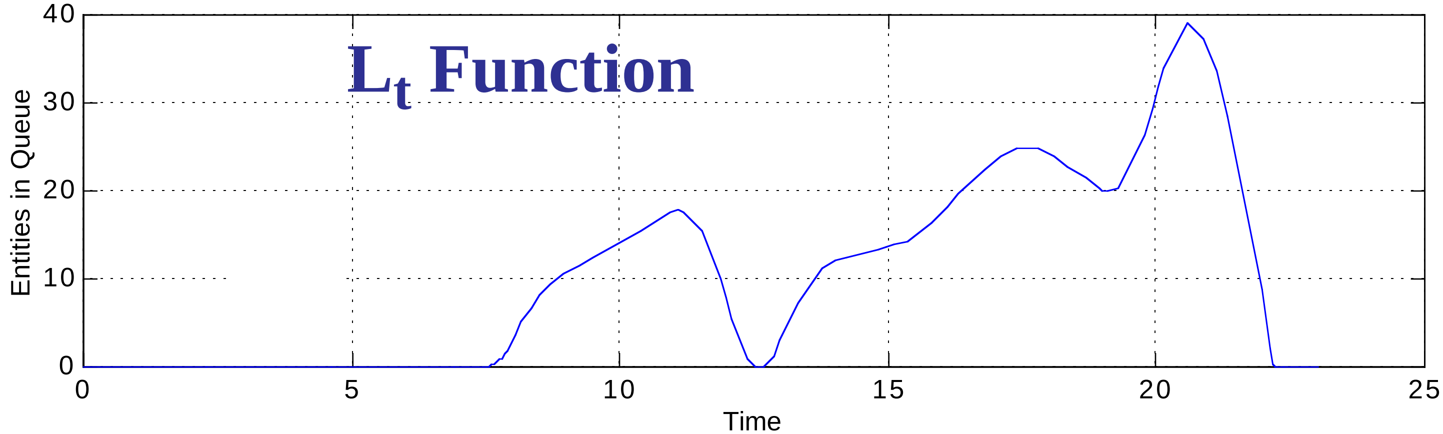
```
% Function to compute rates of change of demand and supply functions
%
% two state variables in the system
% p(1) = L(t) = L(t-1) + [demand(t) - supply(t)] (dt)          % queue length
% p(2) = A(t) = A(t-1) + [L(t)] (dt)                            % area under L(t) curve

function pprime = fqueue_2(t,p)
global demand capacity time
% Define the rate equations
demand_table = interp1(time,demand,t);
capacity_table = capacity;

if (demand_table < capacity_table) & (p(1) > 0)
```

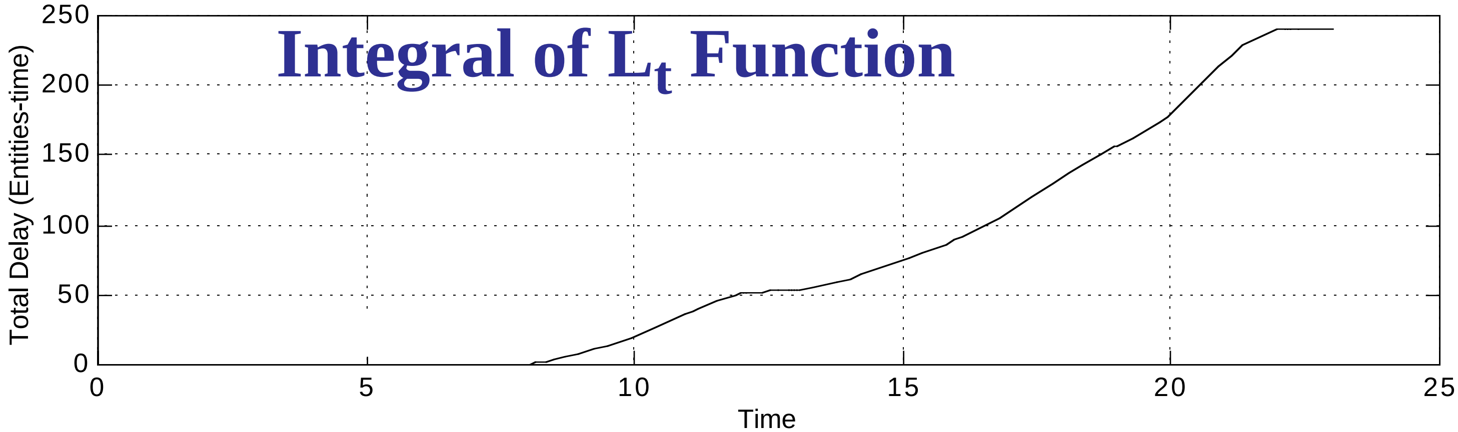
```
pprime(1) = demand_table - capacity_table;    % rate of change in state variable
elseif demand_table > capacity_table
pprime(1) = demand_table - capacity_table;  % rate of change in state variable
else
pprime(1) = 0.0;       % avoids accumulation of entities if queue length is zero
end

pprime(2) = p(1);    % integrates the delay curve over time
pprime = pprime';
```

# Sample Output (Matlab Program)



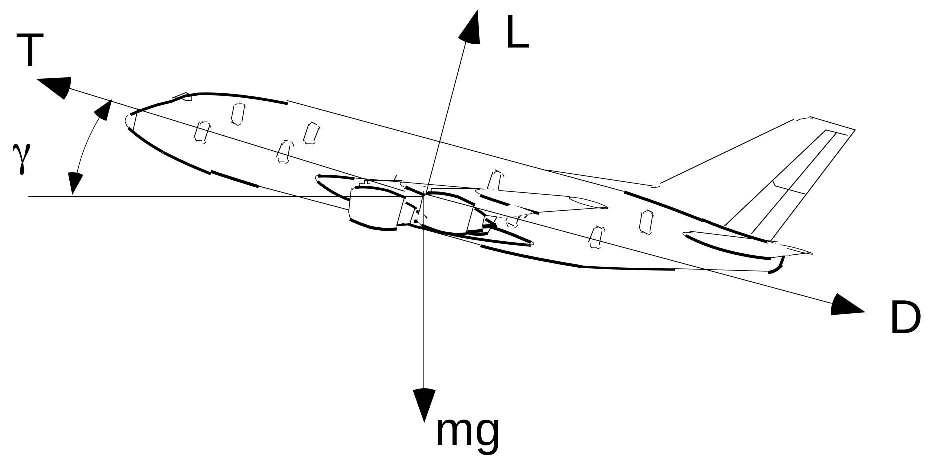
# Sample Results (Queue Length and Area under the Queue Length Curve)



# Example 6 - Aircraft Trajectory Simulation (Climb Performance)

Many airport and airspace simulation models employ simplified algorithms to estimate aircraft climb performance in the terminal area.



# Basic Climb Performance Analysis

The basic equations of motion along the climbing flight path and normal to the flight path of an air vehicle are:

$$m\frac{dV}{dt} = T - D - mg\sin\gamma \qquad (8)$$

$$m\frac{d\gamma}{dt}V = L - mg\cos\gamma \qquad (9)$$

where: m is the vehicle mass, $V$ is the airspeed, T and D are the tractive and drag forces, respectively; $\gamma$ is the flight path angle. L is the lift force and $mg\cos\gamma$ is the gravitational component normal to the flight path.

# Climb Performance Model Simplifications

For small γ (flight path angle):

$$\sin\gamma = \frac{T-D}{mg} - \frac{1}{g}\frac{dV}{dt} \qquad (10)$$

where: the first term in the RHS accounts for possible changes in the potential state of the vehicle (i.e., climb ability) and the second terms is the acceleration capability of the aircraft while climbing. Further algebraic manipulation yields,

$$V\sin\gamma = \frac{dh}{dt} = \frac{V[T-D]}{mg} - \frac{V}{g}\frac{dV}{dt} \qquad (11)$$

where: $dh/dt$ is the rate of climb and V is the airspeed. Note that if one neglects the second term (acceleration factor) assuming small changes in V as the vehicle climbs one can easily estimate the rate of the climb of the vehicle for a prescribed climb schedule.

# Incorporation of a Parabolic Drag Polar Model

Let lift and drag be expressed in the simple parabolic form,

$$L = \frac{1}{2}\rho S C_L V^2 \tag{12}$$

$$D = \frac{1}{2}\rho S C_D V^2 \tag{13}$$

where: $C_L$, $C_D$ are the lift and drag coefficients (nondimensional), V is the airspeed, S is the wing area (reference area) and $\rho$ is the density of the air surrounding the vehicle.

# Final Derivation of Climb Rate Expression

The functional form of the lift and drag coefficients ($C_L$, $C_D$) in its simplest form is,

$$C_D = C_{D0} + C_{Di} = C_{D0} + \frac{C_L^2}{\pi A R e} \qquad (14)$$

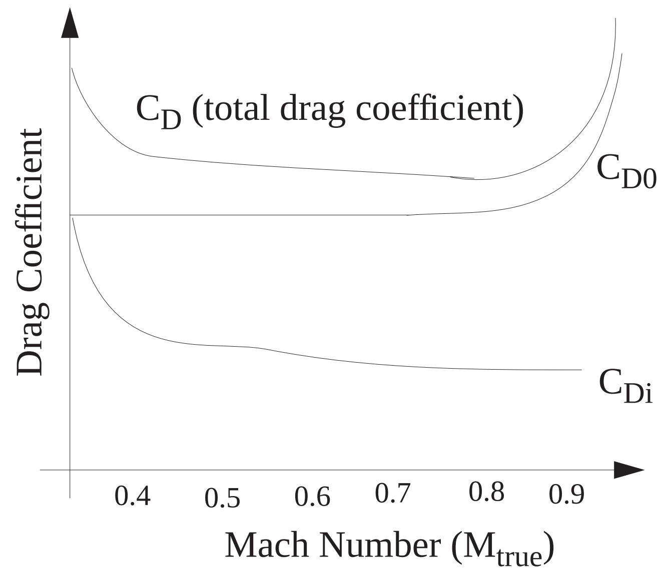
$$C_L = \frac{2mg}{\rho S V^2} \qquad (15)$$

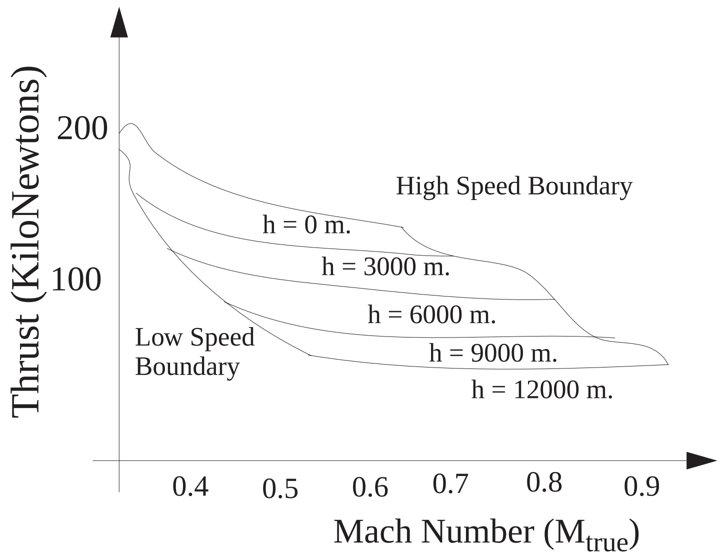
where: $C_{D0}$ is the zero lift drag coefficient, and the second drag term accounts for drag due to lift generation (i.e., induced drag).Then the rate of climb function becomes,

$$\frac{dh}{dt} = \frac{V\left[T(\rho, V) - \frac{1}{2}\rho V^2 S\left\{C_{D0}(M) + \frac{C_L^2(M, V)}{\pi A R e}\right\}\right]}{mg} \qquad (16)$$

# Mathematical Approximation for Aircraft Thrust and Drag

**Thrust and drag are two fundamental variables extracted from wind-tunnel and flight tests.**

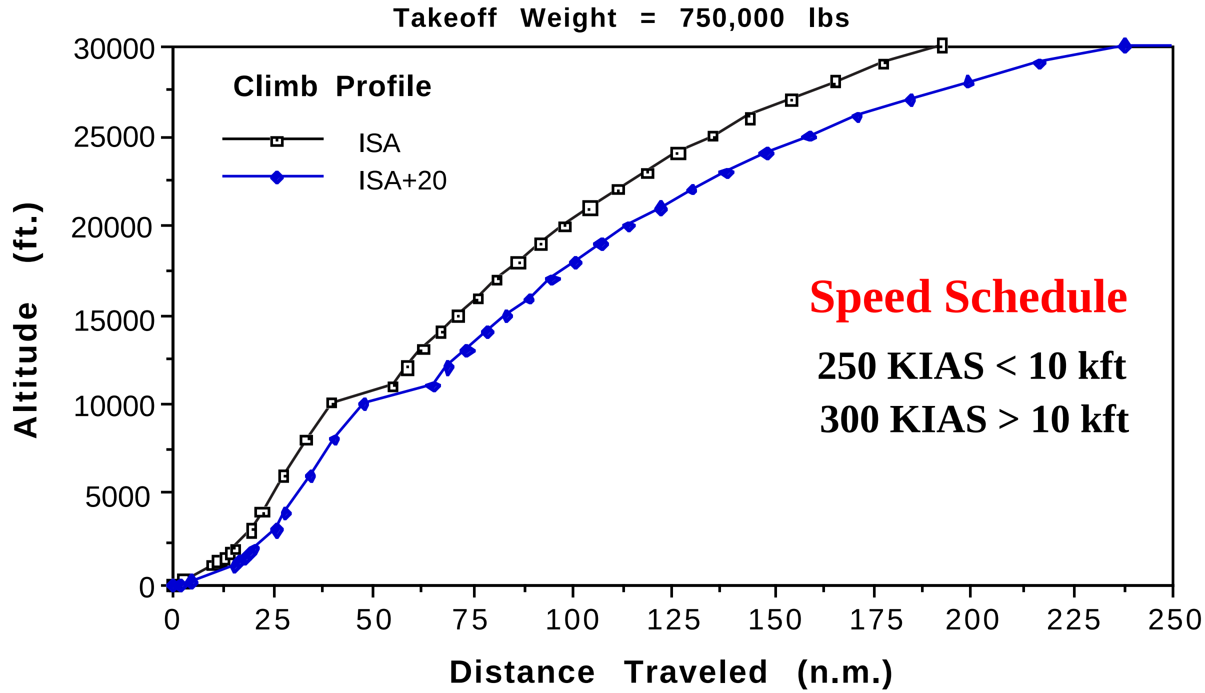


# Sample Climb Trajectory Results

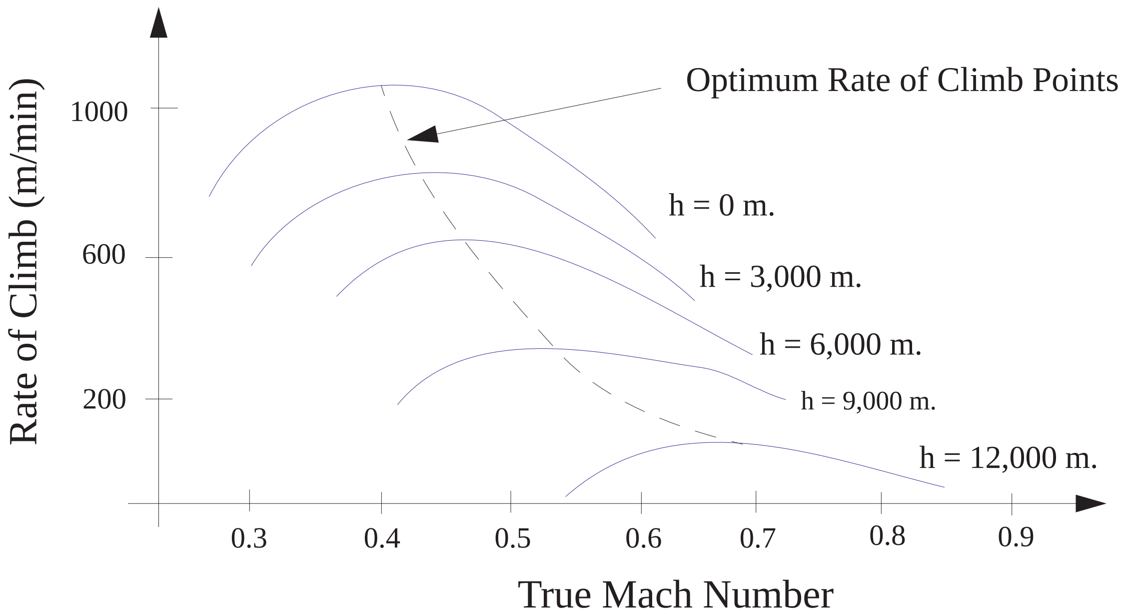
**Numerical integration of equation (9) for a given flight speed schedule (speed time history) yields the following climb profiles.**

Four engine, turbofan-powered aircraft



# Typical Rate of Climb Envelope

Iterative analysis of the rate of climb equation yields the following results across the complete flight envelope.



# Continuous Simulation Languages

ACSL - advanced continuous simulation language

DYNAMO - systems dynamics specific language

STELLA - graphical language with good capabilities

CSMP- IBM precursor of ACSL

SIMSCRIPT II.5 - mainly used for discrete systems

MODSIM - object-oriented simulation language

SLAM III - mainly used for discrete systems

# Remarks About Continuous Simulation

Pros:

- Good causality between variables affecting the system behavior
- Handles time varying stochastic and deterministic processes
- Provides good insight about the dynamics of the system

Cons:

- Require some tool or algorithm to solve the differential equations of the system
- Could be computational intensive

# Discrete Event Simulation

- Description of a system using logical relationships detailing how state variables change over time (discrete changes)
- The most widely used tool when problems are complex in nature
- Heavily used in airport and airspace simulations (SIMMOD, TAAM, RAMS, etc.)
- The basic idea is to move entities in the simulation according to logical constraints and keeping track of the time-space positions for each entity
- Statistics are collected as average times in each process

# Principles of Discrete Event Simulation

Two modeling approaches of discrete event simulation are generally implemented in large-scale airport/ airspace simulations:

## a) Event-scheduling simulation

Estimation of state variables at times when each event occurs (characteristic events are explicitly stated in the simulation process)

## b) Process or activity-based simulation

Description of processes (i.e., time-ordered sequences of events) to model activities “experienced” by each entity throughout the simulation

# Elements of a Simulation Model (Kelton et al.)

Entities: dynamic objects that are created and destroyed as the simulation moves forward in time

Attributes: characteristics of entities used to differentiate behavior in the simulation process (i.e.,transfer vs. terminating passengers)

Global Variables: provide general parameters of a simulation model (i.e., simulation stopping criteria)

Resources: elements that are occupied in the simulation by entities for a finite period of time (i.e., ticket counter occupied by a passenger)

# Elements of a Simulation Model (continuation)

Queues: representation of physical waiting spaces in the simulation model (i.e., uniqueue before a security check point)

Accumulators: variables that keep track of simulation performance measures (i.e., queue length, level of service, etc.)

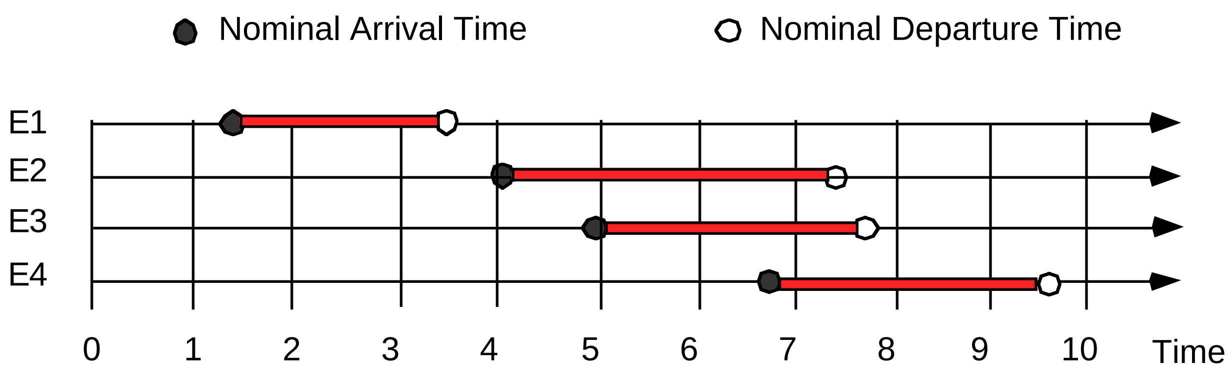
Events: important milestones in the simulation process (i.e., arrivals, departures, simulation initiation and completion). Events can be:

- External (user inputs)

- Internal (the result of entity process conflicts)

# Sample Discrete Event Simulation

Suppose that we have four processes as shown below




| Entity | Arrival Time | Interarrival Time | Service Time |
|---|---|---|---|
| 1 | 1.4 | 2.6 | 2.1 |
| 2 | 4.0 | 0.9 | 3.5 |

| Entity | Arrival Time | Interarrival Time | Service Time |
|---|---|---|---|
| 3 | 4.9 | 1.8 | 2.8 |
| 4 | 6.7 | 0.0 | 2.9 |

# Verbal Description of Events

Assume First-In-First-Out events apply:

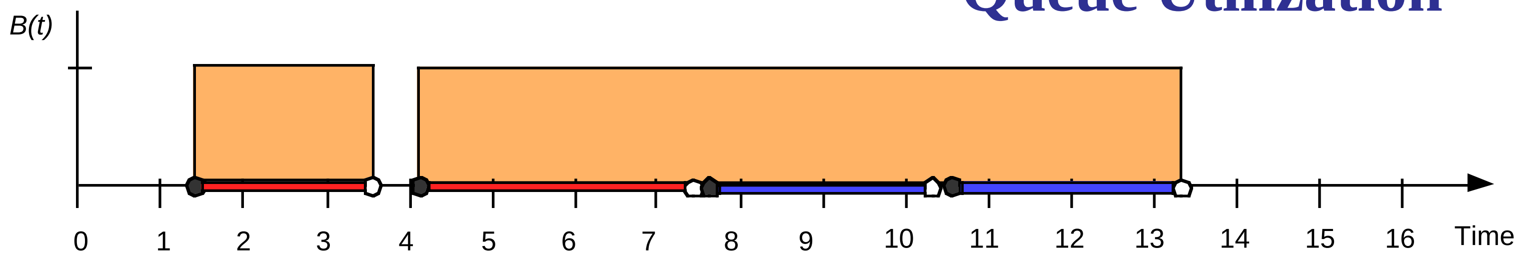
- Entity 1 enters the system and departs without delay
- Entity 2 enters the system and departs without delay
- Entity 3 arrives 0.9 time units after Entity 2 and is delayed 2.6 time units before service (at 7.5 time units)
- Entity 4 arrives at 6.7 time units while Entity 2 is being serviced and Entity 3 waits in the queue
- Entity 4 waits until 10.3 time units to be serviced
- Entity 3 departs at 10.3 time units

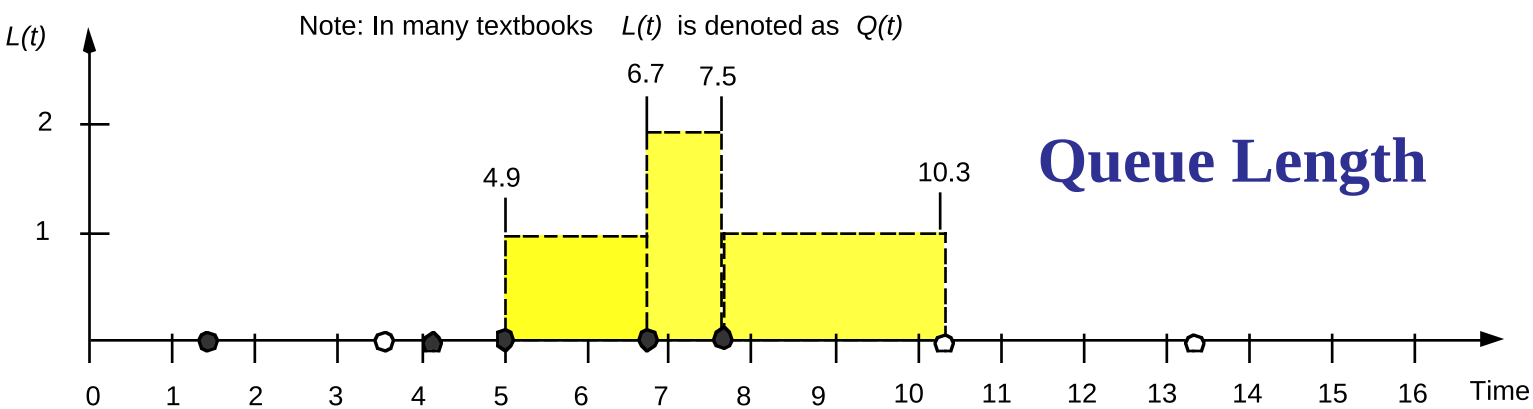
- Entity 4 departs the system at 13.2 time units

# Utilization and Queue Length

The following diagrams represent times when the system is in use and the queue length observed

## Queue Utilization




## Queue Length

## Simulation Parameters

Assume a single resource (server) provides service to these entities. Let,

$D_i$ be the delay for the $i$th entity entering a system

$N$ is the number of entities processed during the simulation time $T$

$L$ is the time-average number of entities in the queue

$Q(t) = L(t)$ is the instantaneous queue length

$B(t)$ is a busy function such that,

$$B(t) = \begin{cases} 1 & \text{if system is busy} \\ 0 & \text{if system is idle} \end{cases}$$

A statistic to compute the average delay of entities processed by the system is,

$$\bar{D} = \frac{\sum_{i=1}^{N} D_i}{N} \qquad (17)$$

Similarly, the time-average number of entities in the queue is computed by integrating $L(t)$ over time (0,T)

$$\bar{L} = \int_0^T L(t)dt / T \quad (18)$$

Finally, the utilization of the system can be deduced from direct observation of the busy function, $B(t)$ and its integral over time (0,T),

$$\rho = \int_0^T B(t)dt / T \quad (19)$$

These metrics constitute the foundations on how a simulation model can be effectively used to predict levels of service inside airport terminals and practically everywhere where a waiting line forms.

# Computations for Hand Calculation Example

Looking back at the previous example we compute the queueing parameters using equations 10-12. Use 16 time units as the simulation life-span.

$$D = (0 + 0 + 2.6 + 3.6)/4 = 1.55 \quad \text{time units}$$

$$\bar{L} = \int_0^{16} L(t)dt/T \approx \sum_{i=1}^{16} L(t)\Delta t/16$$

$$\bar{L} = \frac{(6.7 - 4.9) \times 1 + (7.5 - 6.7) \times 2 + (10.3 - 7.5) \times 1}{16} = 0.525$$

entities

And the utilization factor is,

$$\rho = \int_0^T B(t)dt / T = \sum_{i=1}^{16} B(t)\Delta t / 16 = 0.706$$

From these statistics we conclude the following,

- The use of the single server system is quite good (about 70% of the time the server is busy)
- Simulation is an intensive book keeping activity that is obviously suited to computers
- Simple formulae can be used to obtain vital statistics of the system modeled

# A More Formal Simulation Process

- A more formal simulation process (other than hand calculations) is introduced here
- The example and nomenclature used correspond to that used by Law and Kelton (1991)
- A simple single server queueing process is first explained and a couple of examples are presented
- Results of the simulation process are compared with analytic results derived from a stochastic queueing model

# Simulation Blocks

Initialization: initializes counters (to keep statistics) and global variables

Main: controls other routines and acts as director. Performs calls to others

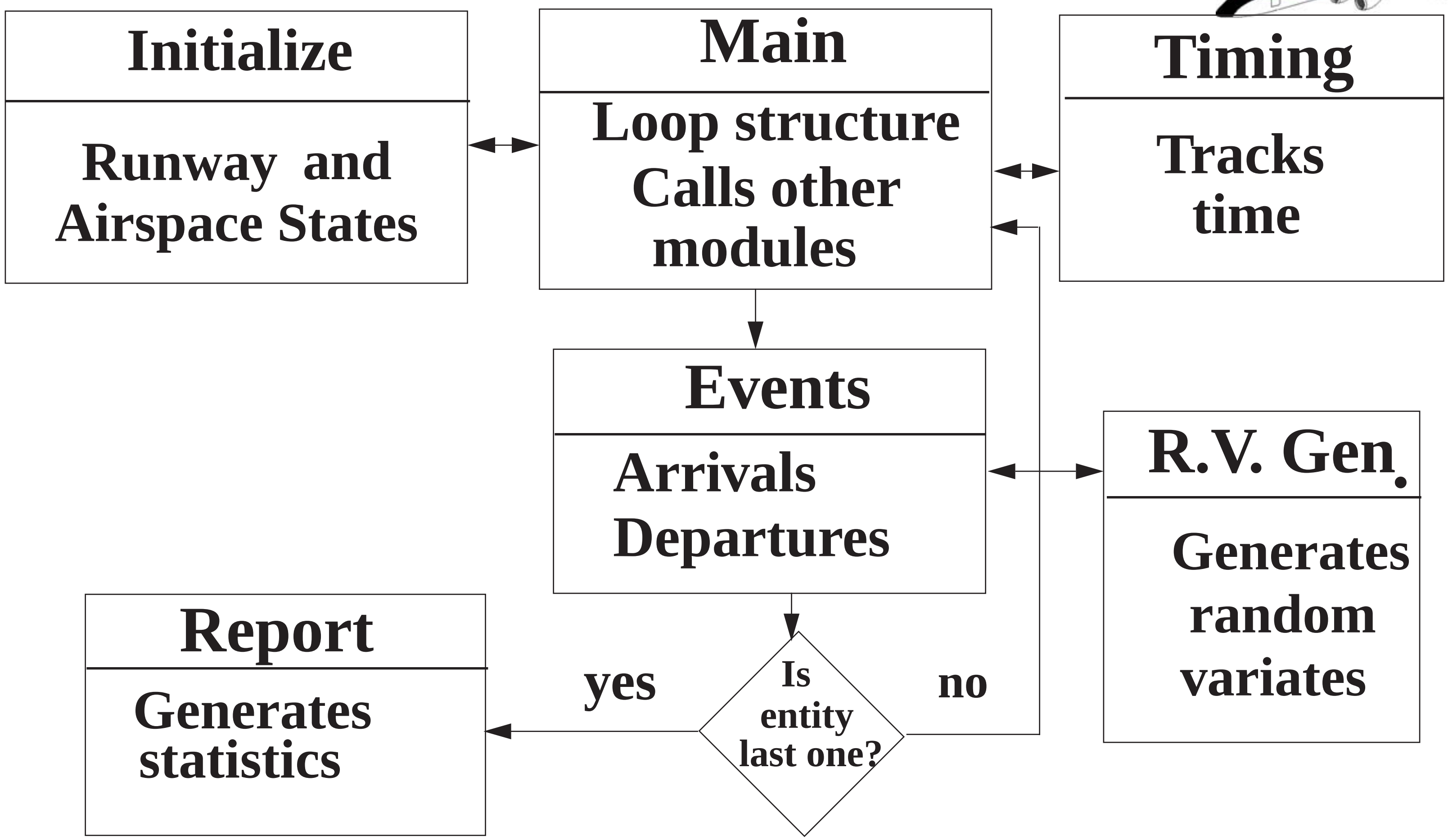
Timing: keeps the simulation clock up-to-date

Report: writes and plots summary statistics of the simulation model

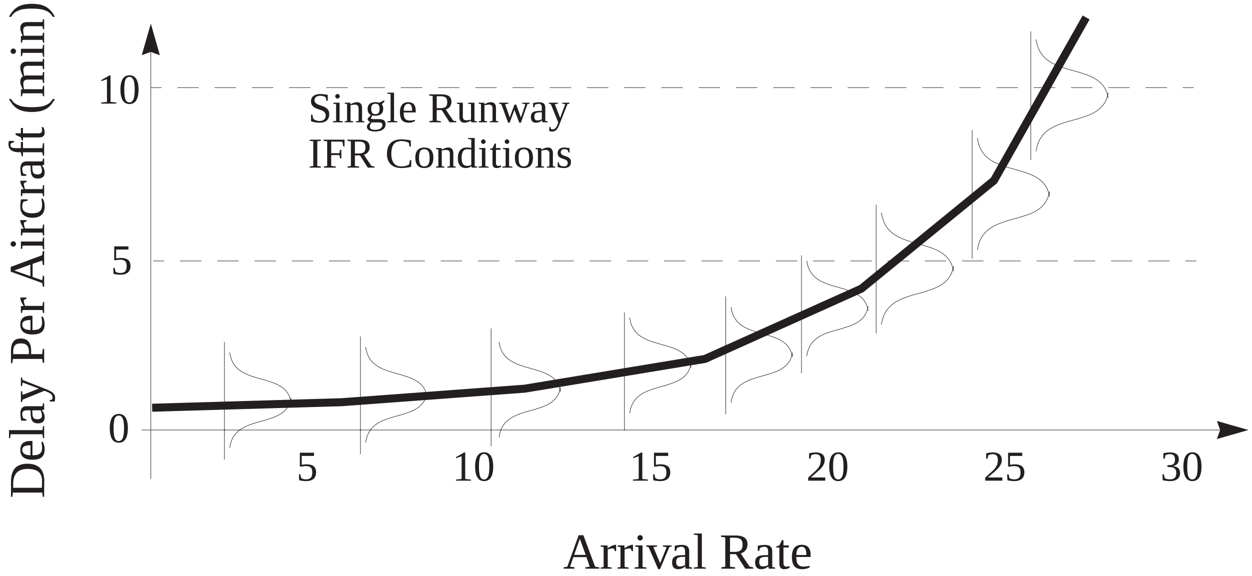
Event: tracks and schedules events in the simulation

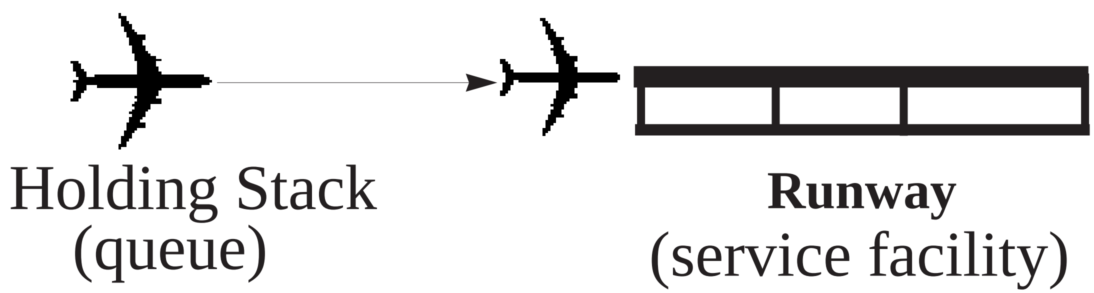
Generator: generates random variates needed in the simulation

# Sample D.E. Simulation (Law and Kelton)



# Typical Results for M/M/1 Queue System



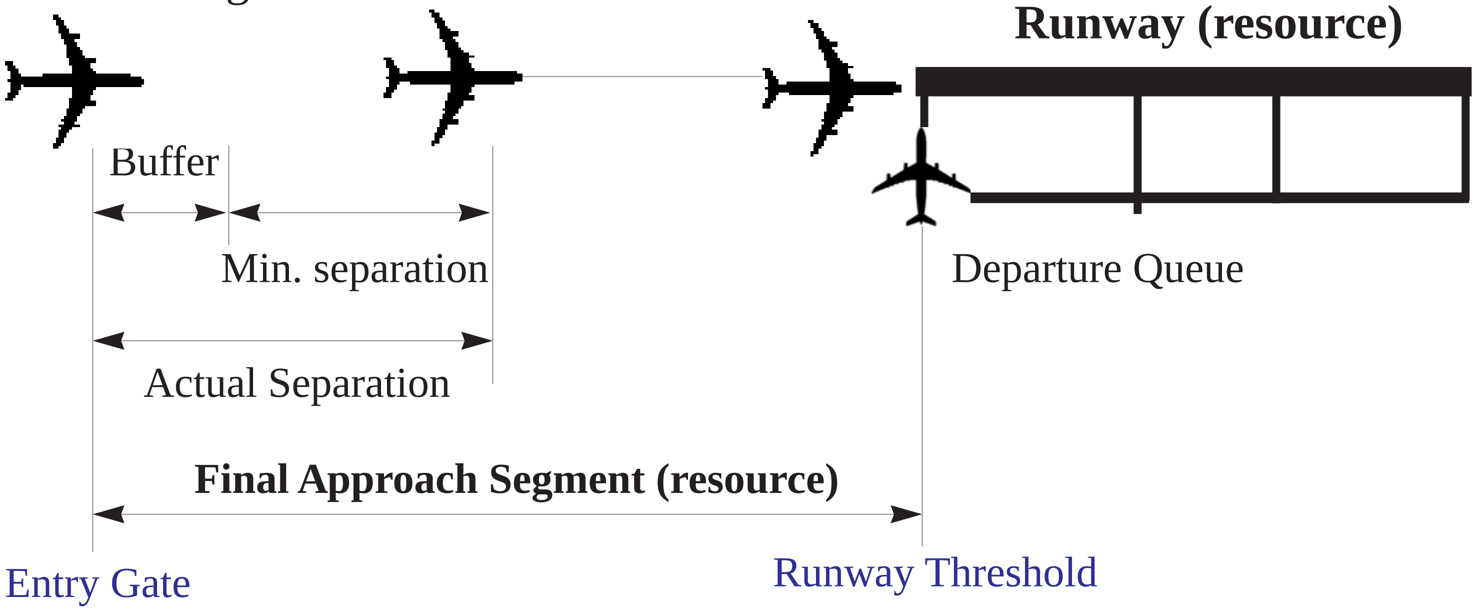



# A More Complex Discrete Event Model

- RUNSIM Model
- Developed at Virginia Tech (Nunna, 1992) to estimate runway occupancy times and exit location use
- SIMSCRIPT II.5 source code
- Considers arrivals (from Final Approach Fix) and departures (runway simulation only)

# RUNSIM Simulation Model (I)

Basic definitions and nomenclature.



# RUNSIM Model Assumptions (I)

1) Arrivals and departures are generated independent of each other using pre-defined arrival distributions (3 types of PDFs).

2) Arrivals are generated at the approach gate.

3) If the arrivals are greater than the processing capacity of the airport system then the arrivals are queued in a stack at the approach gate.

4) Aircraft wake vortex separation rules (including separation buffers) are considered.

5) Arrivals are given priority over departures if both events were to occur at the same time.

6) Aircraft maintain a constant airspeed in the final approach phase.

# RUNSIM Model Assumptions (II)

7) Runway exits are modeled for Runway Occupancy Time (ROT) estimation.

8) The runway has no gradient.

9) Five types of runway exits are modeled explicitly.

10) Flight vehicle parameters (i.e., final approach speed, ROT, etc.) are modeled individually (34 aircraft in the database).

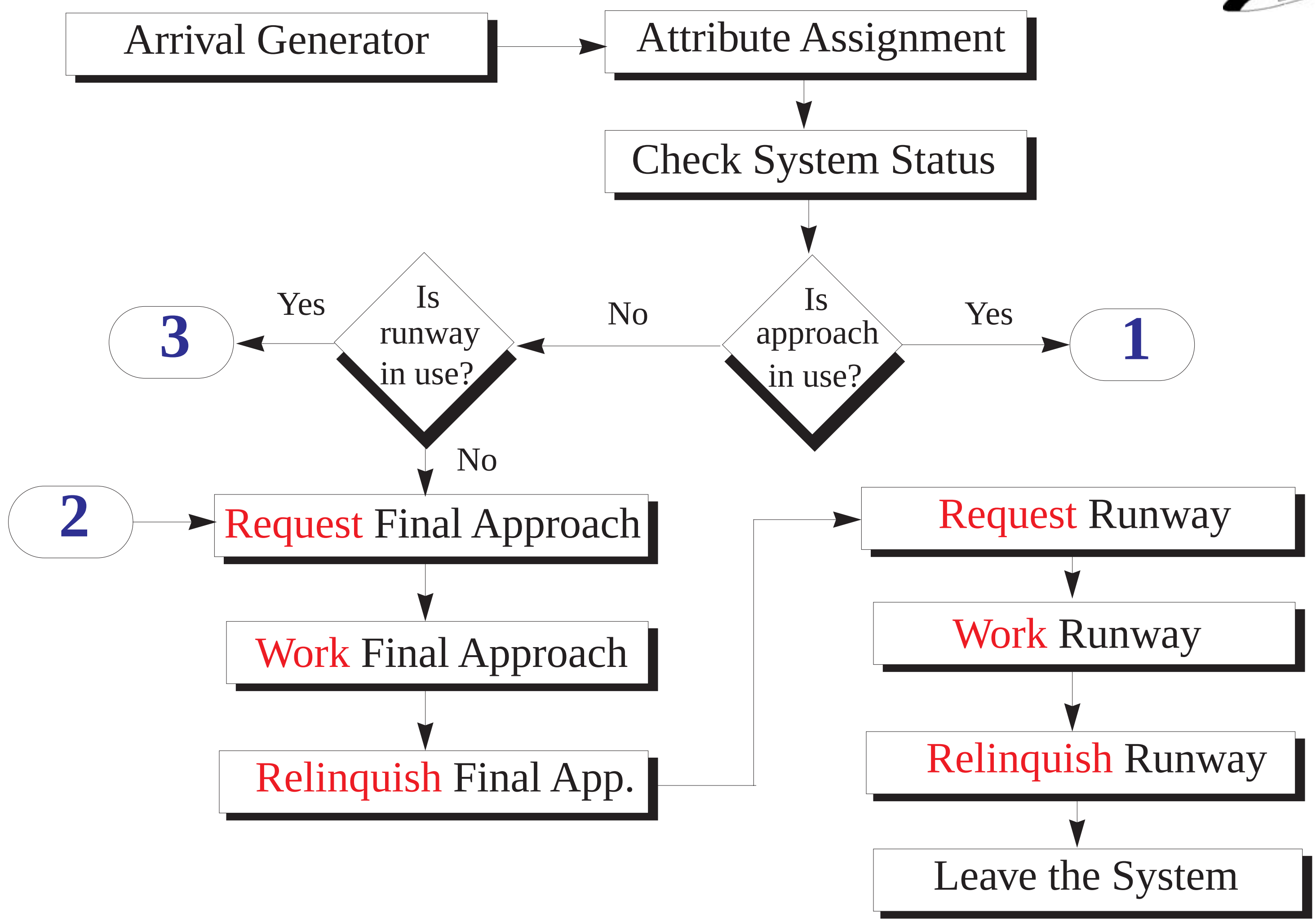
11) User defined aircraft population mix.

12) Airport environmental conditions (i.e., temperature and elevation) are accounted for in the model.

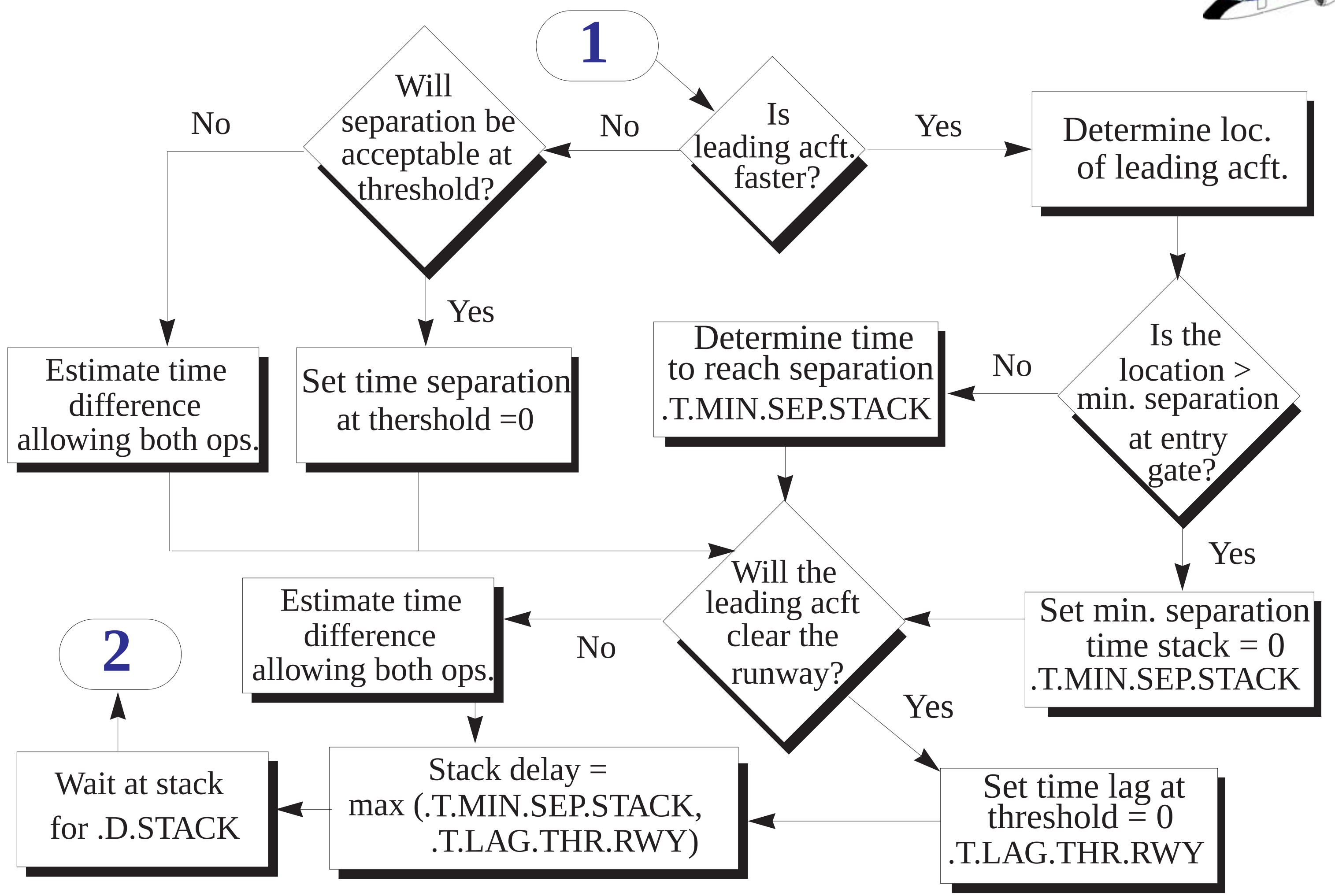
# Modeling Structure (RUNSIM)

| Airport System Element | SIMSCRIPT II.5 Definition | Remarks |
|---|---|---|
| Final approach segment | Resource | Up to 3 aircraft can occupy this resource |
| Runway | Resource | No simultaneous aircraft occupancy is allowed |
| Aircraft | Temporary entities | Speed, size, group, and ROT are entity attributes |
| Aircraft final approach travel | Process | |
| Aircraft landing roll | Process | |
| Aircraft takeoff roll | Process | |
| Departure and arrival queues | Processes | |
| Aircraft arrivals/departures | Events | User selected PDFs |

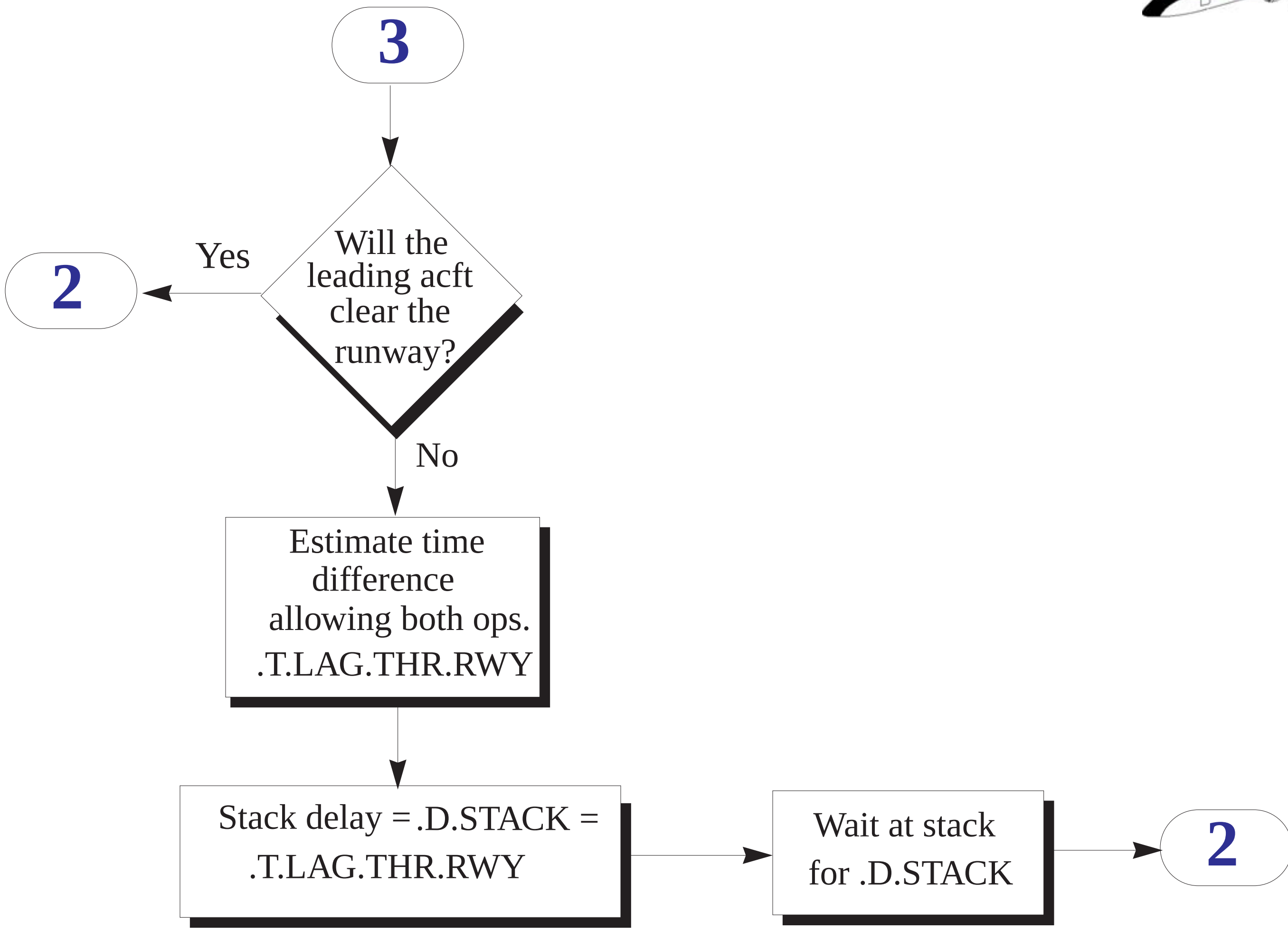
# Arrival Process Flowchart (I)



# Arrival Process Flowchart (II)

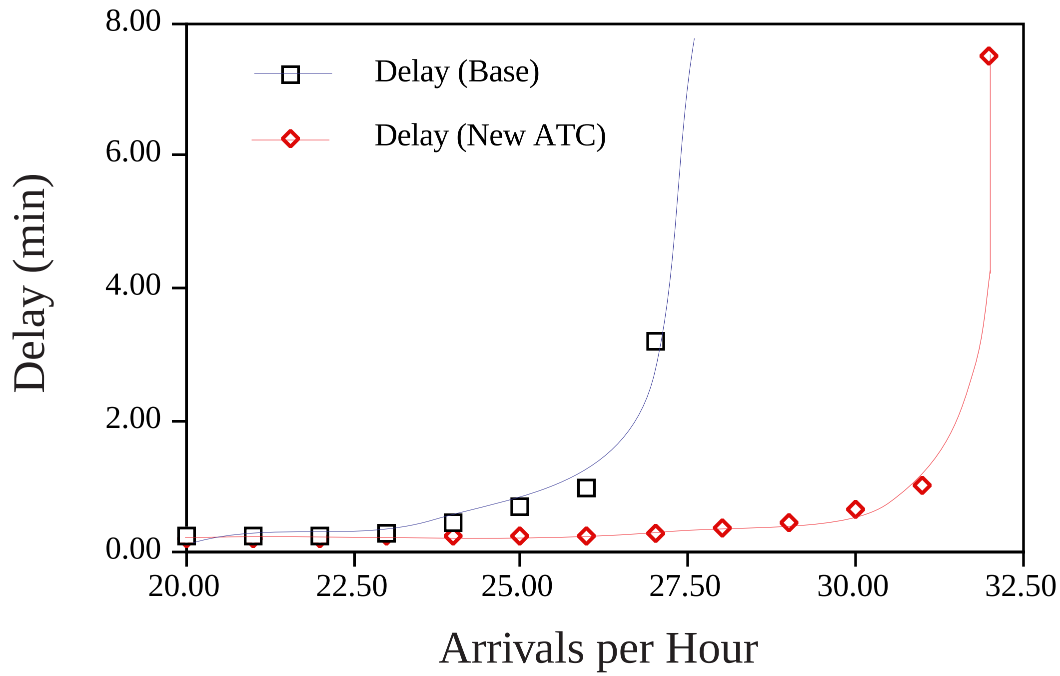


# Arrival Process Flowchart (III)



# Sample Results (RUNSIM)

Delay curves for this single runway airport are illustrated below.



# Discrete Event Simulation Languages

SIMULA

STELLA

SIMSCRIPT II.5

MODSIM - object-oriented simulation language

SLAM III

GPSS-H/GPSS-PC

Many others

# Remarks About Discrete Simulation

Pros:

- Good causality between variables (high fidelity)
- Handles time varying stochastic and deterministic processes
- Provides good insight about the dynamics of the system

Cons:

- Require a simulation language or a good set of simulation libraries
- Very computational intensive
- Validation is expensive due to stochastic nature of the results

# Airport Simulation Models

| Domain of Application | Macroscopic | Microscopic |
|---|---|---|
| Runways | None | REDIM- RUNSIM<br>LMI Run. Model, RDSIM |
| Airfield | Airport Capacity Model (ACM)<br>AND, DELAYS | SIMMOD<br>Airport Machine<br>TAAM |
| Airspace | SIMMOD<br>NASPAC | SIMMOD<br>TAAM<br>RAMS |
| Airport Terminals | APMSIM<br>ALDSIM | ALSIM |

| Domain of Application | Macroscopic | Microscopic |
|---|---|---|
| Airport Noise | INM | SAIC Noise |
| Airport Pollution | EDMS | VPI Plume Model |

# Conclusions on Airport modeling and Simulation Tools

Four methods have been presented to do modeling of airport facilities

Each method represents a level of modeling that should be considered in your project (budget and man-hours)

High fidelity airport simulations can answer specific questions but at a cost.

Several methods could have application in a typical simulation project life cycle.

# References (I)


1) Kulkarni, M., "A Landside Simulation Module Using OOM Discrete Event Simulation Structures", Virginia Tech Thesis, Blacksburg, VA., Fall 1993.

2) Law, A.M. and W.D. Kelton, *Simulation Modeling and Analysis: Second Edition*, McGraw Hill, New York, 1991.

3) Hill, D.R., *Object Oriented Analysis and Simulation*, Addison-Wesley, Harlow, England 1996.

# References (II)


4) Hillier, M. and J. Lieberman, *Introduction to Operations Research: 6th Edition*, McGraw Hill, New York, 1996.

5) Nunna, V.B. “A Computer Simulation Model to Predict Runway Capacity Enhancements”, Virginia Tech Thesis, Blacksburg, VA., Spring 1992.