Electrical & Computer
Engineering Department
**UNIVERSITY OF
PELOPONNESE**

M/M/∞ Queue Simulation

Queuing Theory

Students

Mantvydas Deltuva ece24601

Justinas Teselis ece24600

Professor

Χριστίνα (Τάνια) Πολίτη

Patras, 2024

Contents

Figures

Overview

This report shows an implementation of an **M/M/∞ queueing model** simulation, a foundational model in queueing theory used to study systems with unlimited service capacity.

The M/M/∞ model is defined by:

- **M** (Markovian Inter-Arrival Times): The time between successive arrivals follows an exponential distribution.
- **M** (Markovian Service Times): The service times are exponentially distributed.
- **∞** (Infinite Servers): Every arriving entity is immediately served without delay, as there are unlimited servers available.

The model is commonly used to simulate systems where resources are abundant compared to demand, such as cloud computing or large-scale server farms.

Events in the Simulation

The simulation revolves around two types of **events**:

1. **Arrival Event:** A new call enters the system. It is immediately assigned to a server. Each arrival event generates the next arrival time (*next_arrival_time*) by sampling from the exponential inter-arrival time distribution.
2. **Departure Event:** A call completes its service and leaves the system.

Each event updates the system state, including the current number of active calls (*num_calls*) and the time (*current_time*).

Event Determination Process

At any point, the simulation determines which event (arrival or departure) occurs next:

- **Arrival Time:** Calculated as the current time plus a random inter-arrival time (*exprnd(1 / lambda)*). This is updated every time an arrival event occurs.
- **Departure Time:** The earliest scheduled departure time from the scheduled departures (*scheduled_departure_times*) list.

The simulation compares the next arrival time (*next_arrival_time*) and the earliest departure time (*min(scheduled_departure_times)*) to decide:

- If the arrival time is sooner, an **arrival event** occurs:
  - Updates the number of active calls (*num_calls*).
  - Schedules a departure for the arriving call.
  - Determines the next arrival time by sampling a new inter-arrival time.
- Otherwise, a **departure event** occurs:
  - Reduces the number of active calls (*num_calls*).
  - Removes the completed departure from the schedule.

Simulation Parameters

- **Arrival Rate (λ):** The average number of arrivals per unit time.

```
lambda = 10 % arrivals / unit time
```

- **Service Rate (μ):** The average number of services completed per unit time.

```
mu = 2 % services / unit time
```

- **Total Time:** The duration of the simulation.

```
total_time = 100 % units
```

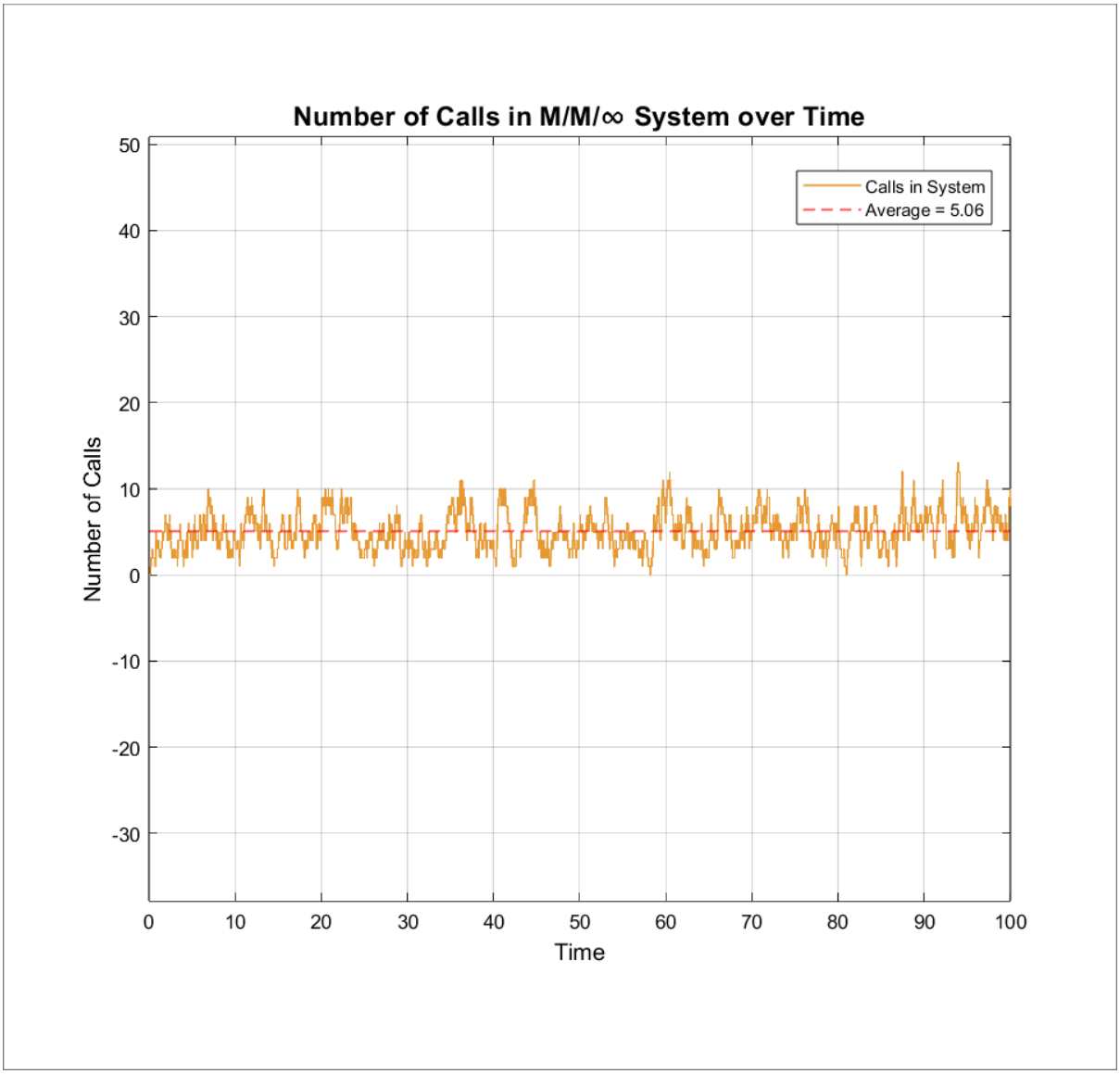
Simulation Flow

1. Start the simulation clock at zero (*current_time = 0*).
2. Generate the first arrival event and initialize the system state:
    - Sample the time until the first arrival from the exponential distribution.
    - Log the initial system state (time and number of calls).
3. Iteratively process events by determining whether the next event is an **arrival** or **departure**:
    - Compare the time of the next arrival with the earliest departure.
    - Update the system state when **Arrival Event** occurs:
        1. Increase the number of active calls.
        2. Schedule a departure for the arriving call by sampling its service time.
        3. Generate the next arrival time by sampling a new inter-arrival time.
    - Update the system state when **Departure Event** occurs:
        1. Decrease the number of active calls.
        2. Remove the completed departure from the schedule.
    - Log the system state after each event (time and number of calls).
4. Continue until simulation time is exceeded (*current_time >= total_time*).

Visualizations

Number of Calls in the System Over Time

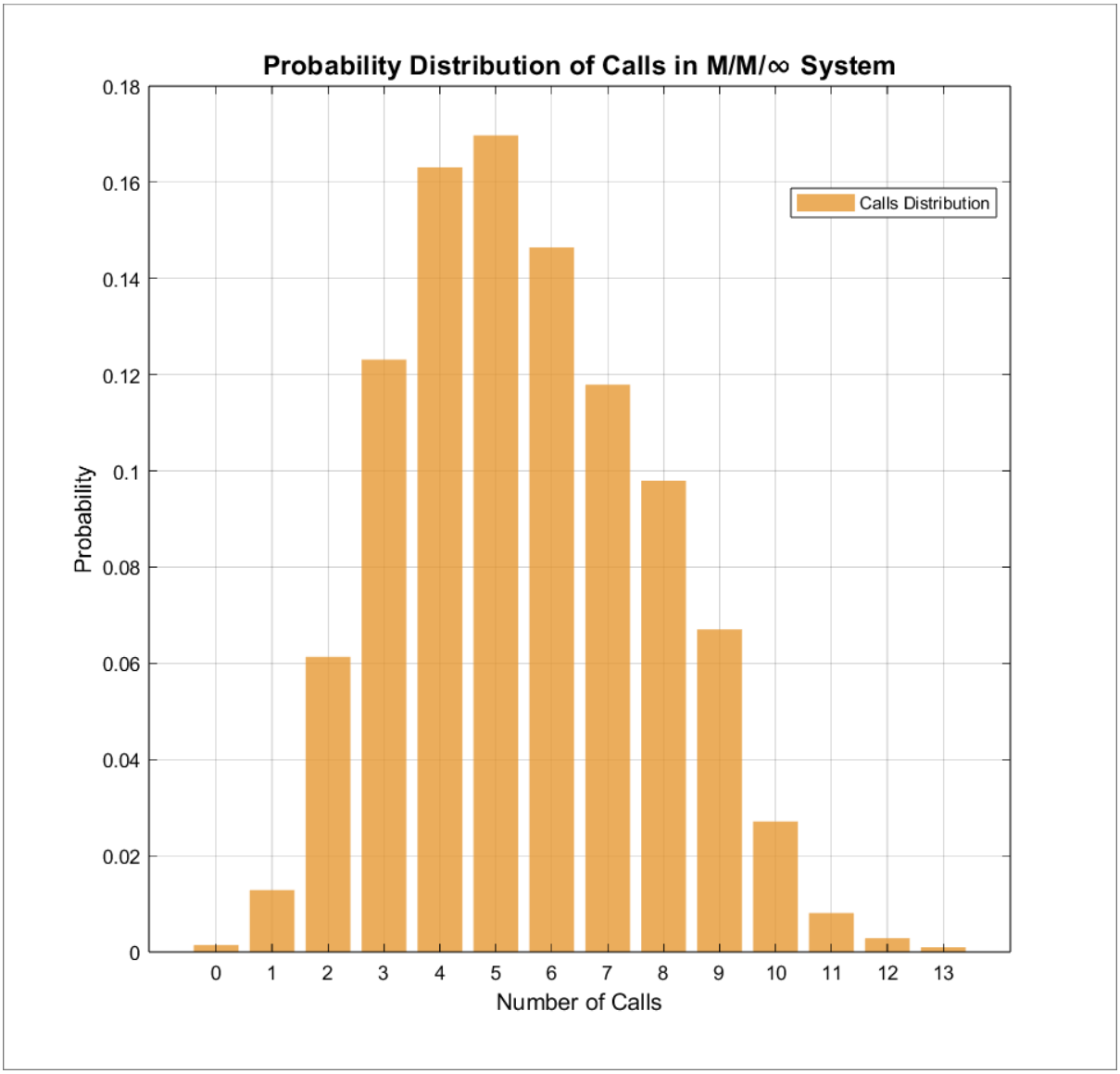
This plot tracks the evolution of the number of active calls during the simulation. The orange line represents the number of active calls at each logged event. The red dashed line shows the time-weighted average number of calls, computed across the simulation duration.



1 pic. Number of Calls in the System Over Time Visualization

Probability Distribution of Calls in the System

This histogram displays the probability distribution of the number of active calls during the simulation. The distribution reflects the system's ability to handle all arriving calls due to infinite servers.



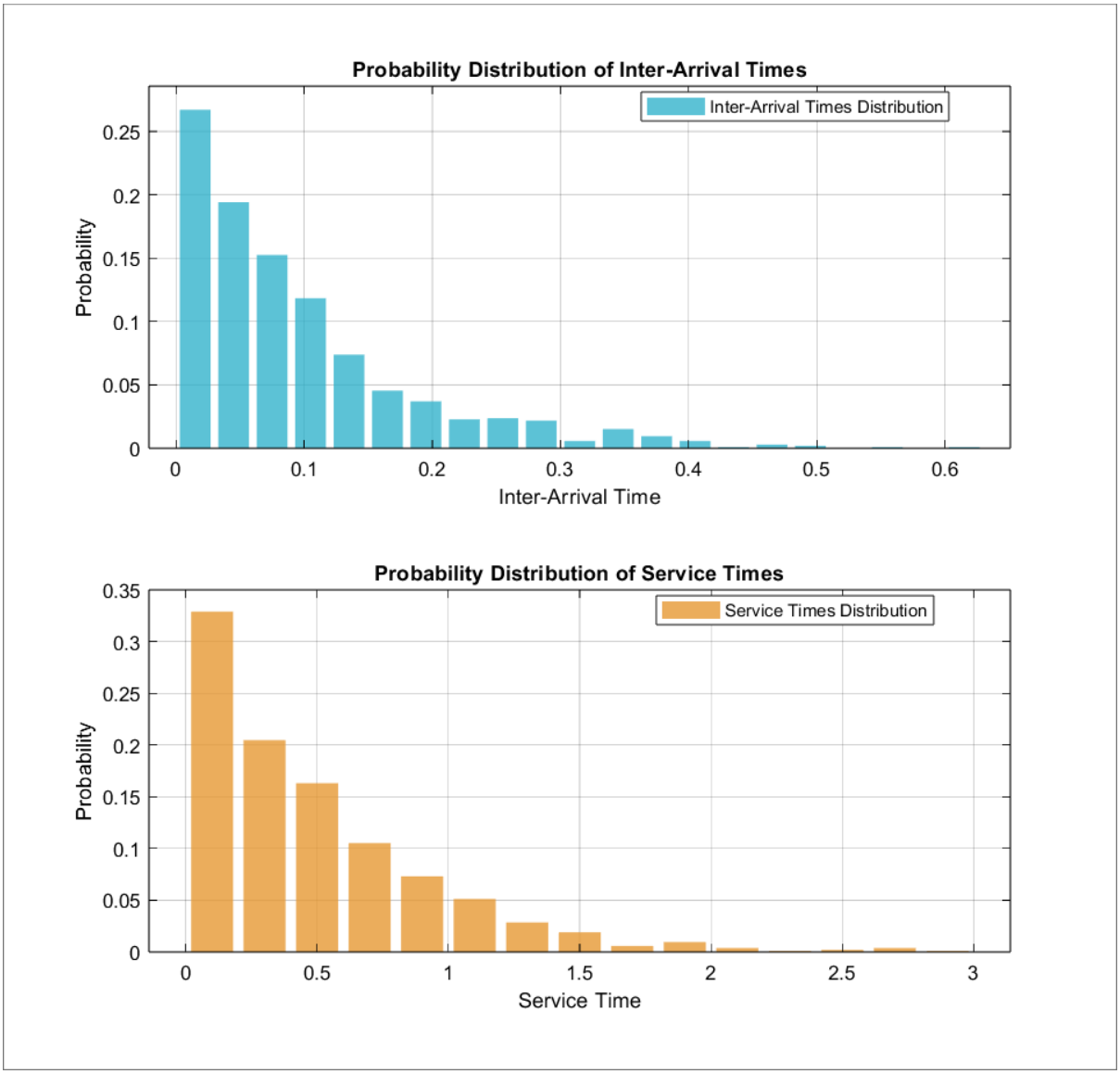
2 pic. Probability Distribution of Calls in the System Visualization

Distributions of Inter-Arrival and Service Times

This plot provides two separate distributions:

1. **Inter-Arrival Times:** The time intervals between consecutive arrivals.
2. **Service Times:** The duration of service for each call.

Both distributions align with the expected exponential behavior, given the model assumptions.



3 pic. Distributions of Inter-Arrival and Service Times Visualization

Source Code

The source code for the implementation of the M/M/∞ queueing model simulation is available on **GitHub**. It includes the full simulation logic, data visualization scripts, and documentation to help users understand and run the simulation with MATLAB. You can access the repository at the following link: M/M/∞ Queue Simulation Source Code.