Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н. Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н. Э. Баумана)

ФАКУЛЬТЕТ «Информатика и системы управления (ИУ)»

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии (ИУ7)»

ОТЧЕТ

по лабораторной работе № 2
по курсу «Моделирование»
на тему: «Марковские процессы»

Студент ИУ7-73Б (Группа) ____________ (Подпись, дата) Р. Р. Хамзина (И. О. Фамилия)

Преподаватель ____________ (Подпись, дата) И. В. Рудаков (И. О. Фамилия)

2022 г.

СОДЕРЖАНИЕ

1 Задание

Реализовать программу с графическим интерфейсом для определения времени пребывания системы в каждом состоянии и предельных вероятностей в установившемся режиме работы системы массового обслуживания. Максимальное количество состояний системы равно десяти.

1.1 Марковский процесс

Случайный процесс, протекающий в некоторой системе S называется марковским, если он обладает следующим свойством: для каждого момента времени вероятность любого состояния системы в будущем зависит только от ее состояния в настоящем и не зависит от того, когда и каким образом система пришла в это состояние.

Для марковского процесса составляют систему уравнений Колмогорова по следующему правилу: в левой части каждого уравнения стоит производная вероятности i-го состояния, в правой части — сумма произведений вероятностей всех состояний, приводящих систему в данное состояние, на интенсивности соответствующих переходов, минус суммарная интенсивность всех переходов, выводящих систему из данного состояния, умноженная на вероятность данного состояния.

1.2 Определение предельных вероятностей

Для определения предельных вероятностей в составленной системе уравнений Колмогорова производные вероятностей состояний заменяются нулевыми значениями, и одно из уравнений заменяется уравнением нормировки: $\sum_{i=1}^{n} p_i(t) = 1$, где n — количество состояний системы.

1.3 Определение точек стабилизации

Для определения точек стабилизации системы определяются вероятности состояний в моменты времени t с малым шагом Δt. Для точки стабилизации должно выполняться условие:

$$|\lim_{t\to\infty} p_i(t) - p_i(t)| < \varepsilon, \text{где } \varepsilon \text{ - заданная точность.} \tag{1.1}$$

2 Реализация

2.1 Детали реализации

На листинге 2.1 показаны реализации функции определения коэффициентов уравнений Колмогорова и функции определения предельных вероятностей.

Листинг 2.1 – Определение коэффициентов уравнений Колмогорова и предельных вероятностей

```
def __get_factors(self):
    factors = []

    for state in range(self.number_states):
        factors.append([0] * self.number_states)

        for i in range(self.number_states):
            if i != state:
                factors[state][i] =
                   self.intensity_matrix[i][state]
                factors[state][state] +=
                   self.intensity_matrix[state][i]
        factors[state][state] *= -1

    return factors

def get_limit_probabilities(self):
    factors = self.__get_factors()
    factors[-1] = [1] * self.number_states

    free_numbers = [0] * self.number_states
    free_numbers[-1] = 1

    return linalg.solve(factors, free_numbers)
```

На листинге 2.2 представлены реализации функции определения производных и функции определения точек стабилизации.

Листинг 2.2 – Определение производных и точек стабилизации

```
def __get_derivatives(self, probabilities, time, factors):
    derivatives = [0] * self.number_states

    for state in range(self.number_states):
```

```python
        for i, probability in enumerate(probabilities):
            derivatives[state] += factors[state][i] * probability

    return derivatives

def get_stabilization_time(self, limit_probabilities):
    time = arange(0, Constants.max_time, Constants.time_delta)

    start_probabilities = [0] * self.number_states
    start_probabilities[0] = 1

    factors = self.__get_factors()

    integrated_probabilities =
        transpose(odeint(self.__get_derivatives,
                         start_probabilities,
                         time, args=(factors,)))

    stabilization_time = []

    for state in range(self.number_states):
        probabilities = integrated_probabilities[state]

        for i, probability in enumerate(probabilities):
            if abs(limit_probabilities[state] - probability) < Constants.eps:
                stabilization_time.append(time[i])
                break

            if i == len(probabilities) - 1:
                stabilization_time.append(0)

    return stabilization_time
```

2.2 Полученный результат

На рисунках 2.1-2.2 и 2.3-2.4 представлены страницы программы для определения времени пребывания системы в каждом состоянии и предельных вероятностей в установившемся режиме работы систем массового обслуживания, состоящих из четырех и пяти состояний соответственно.

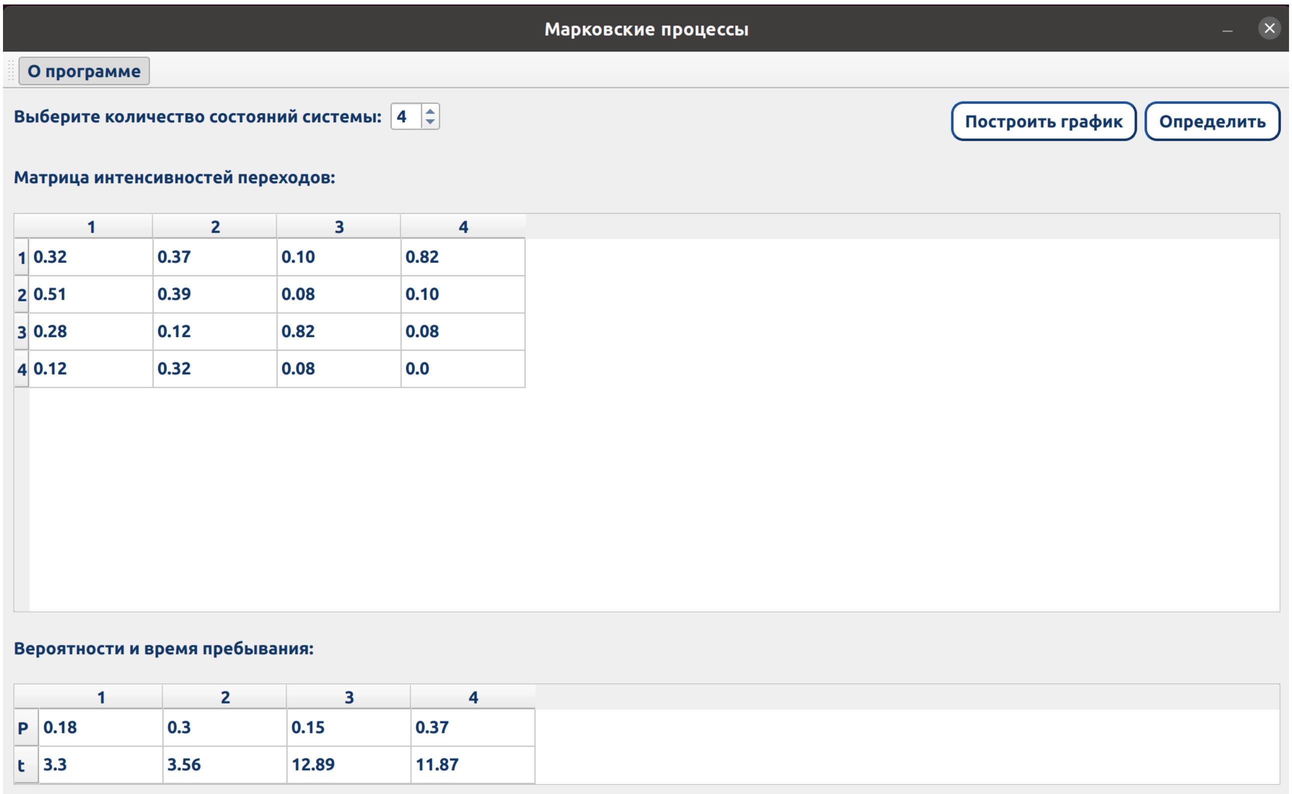

Рисунок 2.1 – Система массового обслуживания, состоящая из четырех состояний

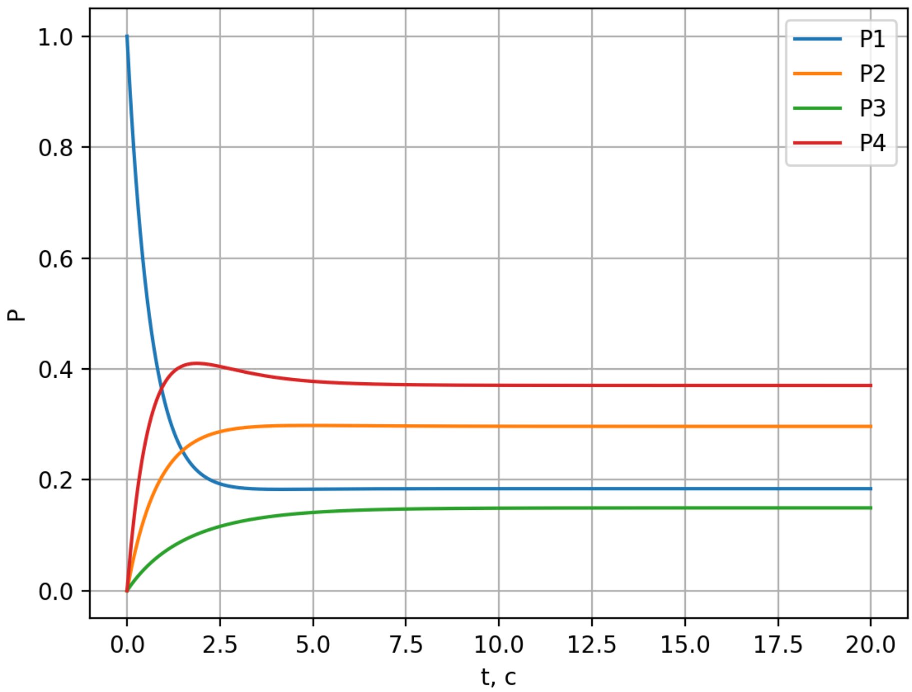

Рисунок 2.2 – График зависимости вероятности от времени для системы, состоящей из четырех состояний

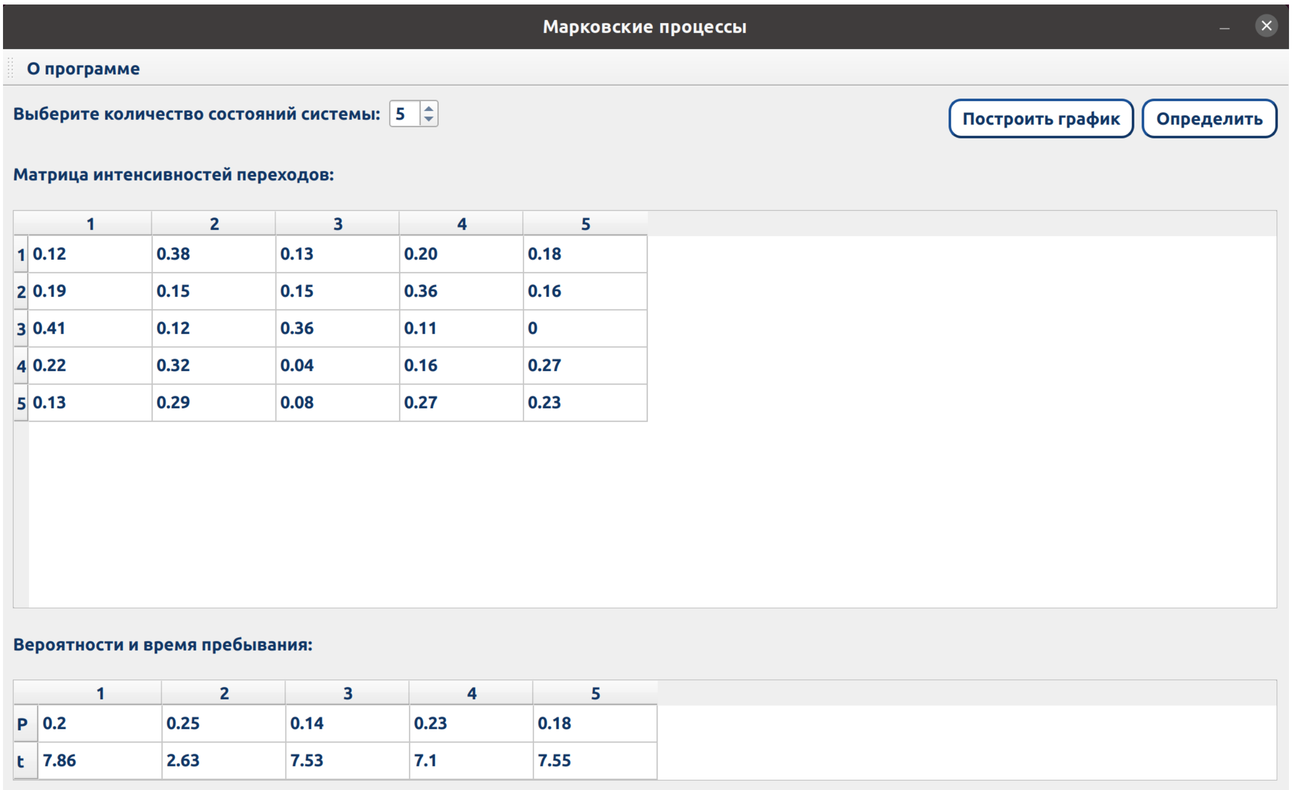

Рисунок 2.3 – Система массового обслуживания, состоящая из пяти состояний

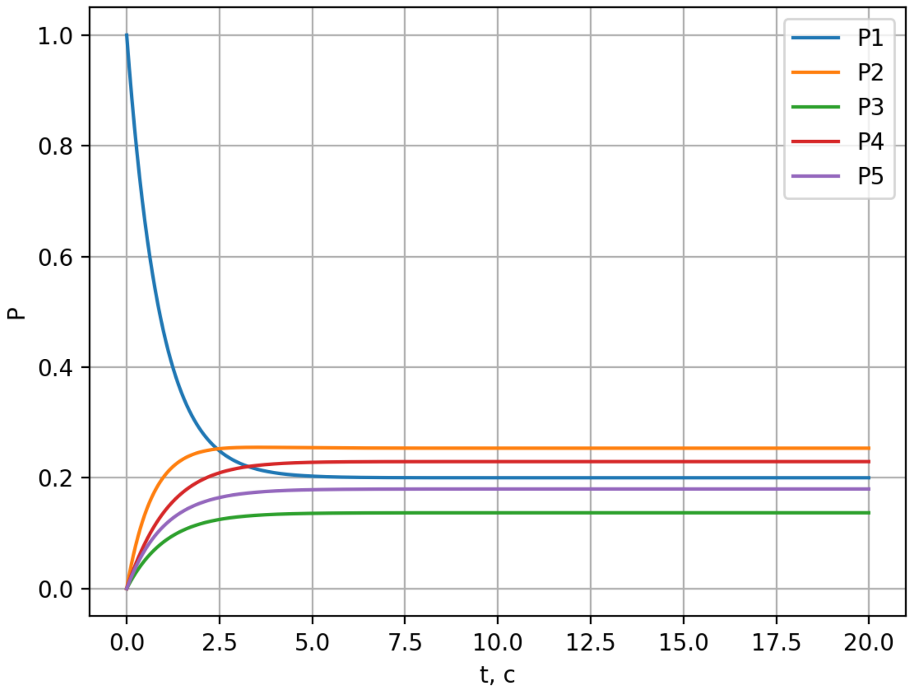

Рисунок 2.4 – График зависимости вероятности от времени для системы, состоящей из пяти состояний