



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»

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

Отчёт по лабораторной работе №2 по курсу «Моделирование»

Тема Марковские процессы

Студент Калашков П. А.

Группа ИУ7-76Б

Оценка (баллы) _____

Преподаватели Рудаков И. В.

Целью данной работы является разработка программы с графическим интерфейсом для определения среднего относительного времени пребывания системы в предельном стационарном состоянии, а также вероятности нахождения системы в указанных состояниях и время до стабилизации вероятностей состояний. Интенсивности переходов из состояния в состояние задаются в виде матрицы (предусмотреть определение размера матрицы).

Марковский процесс

Случайный процесс называется марковским случайным процессом, если для каждого момента времени вероятность люого состояния системы в будущем зависит только от состояния системы в настоящем и не завписит от того, когда и каким образом система пришла в это состояние.

Для марковского случайного процесса составляют уравнения Колмогорова, следуя следующему правилу: в левой части каждого уравнения находится производная функции, отражающей вероятность нахождения системы в i -ом состоянии, в правой части находится столько членов, сколько трелок связано с данным состоянием в направленном графе состояний, причём если стрелка выходит из состояния, член имеет знак минус, если в состояние, знак плюс.

Таким образом, в правой части находится сумма произведений вероятностей всех состояний, переводящих систему в данное состояние, на интенсивности соответствующих переходов, минус суммарная интенсивность всех переходов, выводящих систему из данного состояния, умноженная на вероятность данного состояния. Уравнение Колмогорова для состояния с номером i будет иметь следующий вид:

$$p_i'(t) = \sum_{j=1}^n \lambda_{ji} p_j(t) - p_i(t) \cdot \sum_{j=1}^n \lambda_{ij}, \quad (1)$$

где:

n — число состояний рассматриваемой ситемы;

λ_{ij} — интенсивность перехода системы из i -го состояния в j -ое.

Определение предельных вероятностей

Предельная вероятность состояния — среднее относительное время нахождения системы в данном состоянии. Для определения предельных вероятностей необходимо решить систему уравнений Колмогорова. Поскольку по условию задачи рассматриваемый марковский процесс является стационарным, производные вероятностей заменяются нулями. При этом одно из уравнений в системе необходимо заменить уравнением нормировки: $\sum_{i=1}^n p_i(t) = 1$, где n — количество состояний системы.

Время стабилизации вероятностей состояний

Время стабилизации вероятности нахождения системы в состоянии определяется следующим образом: определяются вероятности нахождения системы в состоянии i в момент времени t и в момент времени $t + \Delta t$ с увеличением значения t до тех пор, пока не будет выполняться следующее условие:

$$|p_i(t) - p_i(t + \Delta t)| < \varepsilon,$$

где ε — заданная точность.

Результаты работы

Детали реализации

В листинге 1 представлена реализация функции получения коэффициентов уравнений Колмогорова, в листинге 2 реализация функции получения предельных вероятностей, а в листингах 3 и 4 реализация функции получения среднего относительного времени.

Листинг 1 – Реализация функции получения коэффициентов уравнений Колмогорова

```
1 def __get_kolmogorov_koeffs(self):
2     factors = []
3
4     for state in range(self.number_states):
5         factors.append([0] * self.number_states)
6         for i in range(self.number_states):
7             if i != state:
8                 factors[state][i] = self.intensity_matrix[i][state]
9                 factors[state][state] -=
10                     self.intensity_matrix[state][i]
11
12     return factors
```

Листинг 2 – Реализация функции получения предельных вероятностей

```
1 def get_limit_probabilities(self):
2     coeffs = self.__get_kolmogorov_koeffs()
3     coeffs[0] = [1] * self.number_states
4     free_numbers = [0] * self.number_states
5     free_numbers[0] = 1
6
7     return linalg.solve(coeffs, free_numbers)
```

Листинг 3 – Реализация функции получения среднего относительного времени (часть 1)

```
1 def get_time_to_stable(self, probs):
2     time = arange(0, max_time, time_delta)
3
4     start_probabilities = [0] * self.number_states
5     start_probabilities[0] = 1
6
7     coeffs = self.__get_kolmogorov_koeffs()
8
9     integrated_probabilities =
10         transpose(odeint(self.__get_derivatives,
11             start_probabilities,
12             time, args=(coeffs)))
```

Листинг 4 – Реализация функции получения среднего относительного времени (часть 2)

```
1      stabilization_time = []
2
3      for state in range(self.number_states):
4          probabilities = integrated_probabilities[state]
5
6          for i, probability in enumerate(probabilities):
7              if abs(probs[state] - probability) < eps:
8                  stabilization_time.append(time[i])
9                  break
10
11             if i == len(probabilities) - 1:
12                 stabilization_time.append(0)
13
14      return stabilization_time
```

Примеры работы

На рисунках 1 и 2 представлены примеры работы разработанной программы для систем из трёх и пяти состояний.

Количество состояний системы: 3

Определить

Матрица интенсивностей переходов:

	1	2	3
1	3	3	3
2	3	3	3
3	3	3	3

Вероятности и среднее относительное время пребывания в предельном стационарном состоянии:

	1	2	3
p	0.33	0.33	0.33
t	2.37	2.37	2.37

Рисунок 1 – Пример работы программы для системы из трёх состояний

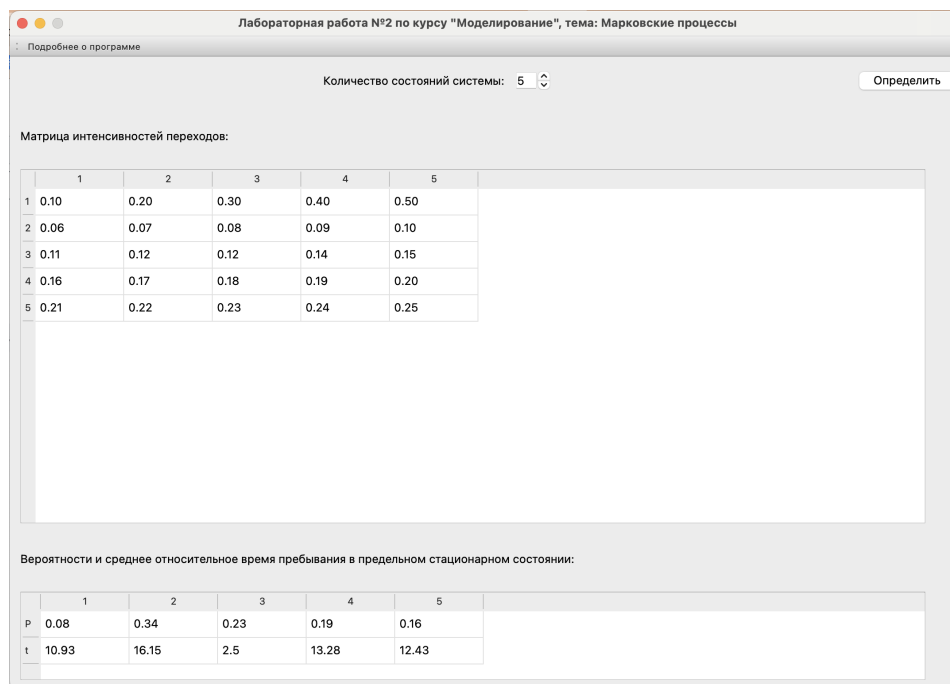


Рисунок 2 – Пример работы программы для системы из пяти состояний

Вывод

В ходе выполнения лабораторной работы была реализована программа с графическим интерфейсом для определения среднего относительного времени пребывания системы в предельном стационарном состоянии, а также вероятности нахождения системы в указанных состояниях.