

Министерство науки и высшего образования Российской Федерации

Рязанский институт (филиал)
федерального государственного бюджетного образовательного учреждения
высшего образования
«Московский политехнический университет»

Кафедра «Информатика и информационные технологии»

Миронова Е.И.

МОДЕЛИРОВАНИЕ СИСТЕМ УПРАВЛЕНИЯ

Учебно-методическое пособие

Рязань

2020

УДК 378
ББК 32.81
М93

Миронова, Е.И.

М93 Моделирование систем управления. Учебно-методическое пособие / Е.И. Миронова – Рязань: Рязанский институт (филиал) Московского политехнического университета, 2020. – 50 с.

Данное пособие предназначено для студентов бакалавриата направлений подготовки 09.03.01 «Информатика и вычислительная техника» и 27.03.04 «Управление в технических системах», изучающих дисциплину «Моделирование систем управления». Данное пособие поможет выполнению курсовой работы, закреплению знаний, полученных на лекциях, лабораторных и практических занятиях, в процессе самостоятельного изучения литературных источников, получат навыки самостоятельной исследовательской работы.

Печатается по решению Учебно-методического совета Рязанского института (филиала) Московского политехнического университета.

УДК 378
ББК 32.81

© Миронова Е.И., 2020

©Рязанский институт (филиал) Московского
политехнического университета, 2020

ОБЩИЕ ВОПРОСЫ

Моделирование является мощным средством анализа и синтеза сложных объектов, процессов и явлений. Решение многих сложных научных и технических задач значительно упрощается при моделировании, т.е. замещении одних объектов другими, обеспечивающими отражение наиболее существенных для исследователя свойств и особенностей замещаемых объектов. В настоящее время моделирование находит широкое применение в автоматике, вычислительной и измерительной технике, радиотехнике и связи, математике и других областях науки и техники.

Моделирование состоит в выявлении основных свойств исследуемого процесса, построении моделей и их применении для прогнозирования поведения натуры. Критерием правильности моделирования является практика.

Ценность методов моделирования состоит в том, что они позволяют существенно сократить и облегчить натурный эксперимент, который обычно дорог, сложен и связан с аварийными состояниями, а также увеличить достоверность математического описания и расчетов. Моделью может являться реальное техническое устройство и абстрактное математическое описание, отображающее натуру. Применение технических средств придает моделированию экспериментальный характер, а модельное математическое описание теоретически раскрывает характер явления.

В настоящее время для динамических исследований систем автоматического управления (САУ) широко используются методы машинного моделирования на основе электронных вычислительных машин: цифровых (ЦВМ), аналоговых (АВМ) и аналого-цифровых комплексов (АЦВК).

С моделированием САУ тесно связана проблема идентификации. Идентификацией динамической системы (процесса) называется получение или уточнение по экспериментальным данным математической модели этой системы или процесса. Под идентификацией в широком смысле понимается получение или уточнение по экспериментальным данным модели реального

объекта, выраженной в тех или иных терминах. Как видно из этого определения, идентификация в широком смысле составляет неотъемлемую часть всякой подлинной науки и имеет древнее происхождение.

В настоящее время в связи с предъявлением все более высоких требований к процессам управления в различных областях техники проблема идентификации становится исключительно важной.

Нельзя обеспечить качественное управление системой, если неизвестна с достаточной точностью ее математическая модель. Для построения математической модели могут быть использованы как теоретические, так и экспериментальные методы. Опыт, накопленный при проектировании систем управления, убедительно свидетельствует о том, что нельзя построить математическую модель, адекватную реальной системе, только на основе теоретических исследований физических процессов в системе. Поэтому в процессе проектирования систем управления одновременно с теоретическими исследованиями проводятся многочисленные эксперименты по определению и уточнению математической модели системы, т.е. ее идентификация.

При решении задачи идентификации в узком смысле считается, что известны структура системы и класс моделей, к которому она относится. Априорная информация о системе достаточно обширна. Такая постановка задачи идентификации наиболее соответствует реальным условиям проектирования и широко используется в инженерной практике.

Цель указаний - ввести студентов специальности «Управление и информатика в технических системах» в теорию и практику моделирования САУ.

1. СТРУКТУРНЫЕ МЕТОДЫ ЦИФРОВОГО МОДЕЛИРОВАНИЯ СИСТЕМ АВТОМАТИЧЕСКОГО УПРАВЛЕНИЯ

1.1. Способы математического описания автоматических систем

Моделирование на ЭВМ – эффективное средство исследования звеньев и систем автоматического управления. На модели можно оценить качество процесса управления автоматической системы при различных возмущающих и задающих воздействиях, а также при изменении характеристик объекта. Это позволяет значительно ускорить сроки проектирования, поскольку в эксплуатационных условиях оценку качества управления при различных режимах работы объекта производить довольно трудно, так как это связано с нарушениями хода технологического процесса и большой затратой времени. Кроме того, при моделировании на ЭВМ появляется возможность наблюдения всех сигналов структурной схемы системы управления.

Успех в решении задачи моделирования САУ на ЭВМ зависит от нескольких основных факторов: адекватности математического описания объекта моделирования, эффективности прикладных методов теории автоматического управления, применяемых вычислительных методов, наличия высококачественного программного обеспечения, степени использования творческого потенциала исследователя. Последний фактор стимулирует развитие диалоговых методов моделирования.

В настоящее время широко используются три основных способа математического описания автоматических систем [1,2]:

- 1) метод передаточных функций и частотных характеристик;
- 2) метод переменных состояния;
- 3) структурно-топологические методы.

В методе передаточных функций для изучения качественного поведения решений обыкновенных дифференциальных уравнений применяются преобразования Лапласа и частотная теория. Достоинством метода является

то, что непосредственное изучение исходных дифференциальных уравнений с заданными начальными условиями заменяется исследованием алгебраических свойств некоторых функций, порождаемых системой дифференциальных уравнений. Многолетние исследования этих методов показали, что по глубине и степени завершенности они не имеют эквивалентной замены, а модификация их для использования на ЭВМ существенно повышает их эффективность.

В основе метода переменных состояния лежит представление модели САУ в виде системы дифференциальных уравнений первого порядка в нормальной форме Коши, которые дополняются алгебраическими уравнениями, связывающими выходные переменные с переменными состояния:

$$\left. \begin{aligned} \dot{x} &= Ax + Bu; \\ y &= Cx, \end{aligned} \right\}$$

где x – вектор переменных состояния размерности $(n \times 1)$; u – вектор воздействий размерности $(m \times 1)$; y – вектор измерений размерности $(r \times 1)$; A , B , C – матрицы коэффициентов размерности $(n \times n)$, $(n \times m)$, $(r \times n)$ соответственно.

Математической основой метода переменных состояния является матричное исчисление и вычислительные методы линейной алгебры.

Структурно-топологические методы опираются на использование теории графов. С точки зрения автоматизации проектирования систем управления, создания диалоговых систем моделирования, использующих в качестве исходной информации структуру системы, эти методы наиболее предпочтительны. Если структура САУ выбрана, то, используя информацию о передаточных функциях отдельных звеньев, способы их соединения и правила эквивалентных преобразований структурных схем, сравнительно легко фор-

мализовать процессы моделирования, как во временной, так и в частотной области [3, 4].

1.2. Структурное цифровое моделирование САУ во временной области

Основным отличительным моментом цифрового моделирования, по сравнению с аналоговым, является использование дискретных форм представления непрерывной информации, позволяющее успешно исследовать с большей точностью количественные и качественные зависимости между элементами сложных динамических систем.

Цифровое моделирование отличается от аналогового тем, что все операции выполняются в цифровой форме с использованием одного из методов численного анализа, со сведением операций интегрирования и дифференцирования к арифметическим операциям. Наиболее простым для реализации является метод Эйлера, согласно которому искомая интегральная кривая заменяется ломаной

$$y_{i+1} = y_i + f(x_i, y_i)h,$$

где h – шаг квантования аргумента.

При h , стремящемся к нулю, ломаная Эйлера (приближение решения) ложится на искомую интегральную кривую. В то же время при уменьшении h увеличивается требуемое для выполнения время. Определение целесообразного значения шага интегрирования может быть произведено по следующей формуле [5, 6]:

$$h = \frac{2\pi}{(12 \div 15)\omega_0} [c],$$

где $\omega_0 = \sqrt[3]{K_{p.c}/T_{min}}$; T_{min} – наименьшая из постоянных времени апериодических звеньев, входящих в состав системы; $K_{p.c}$ – коэффициент передачи разомкнутой системы.

Как известно, динамика любой системы предопределяется структурой последней. Это свойство при структурном моделировании используется следующим образом. Составляется библиотека подпрограмм, включающих цифровые модели типовых динамических звеньев как линейных, так и нелинейных. Разрабатывается специальный модуль, позволяющий в графическом виде вводить структурную схему моделируемой САУ, вид и коэффициенты звеньев. В процессе моделирования на каждом шаге независимой переменной (времени) производится поочередный опрос моделей динамических звеньев для вычисления следующей точки интегральной кривой и переопределение входных сигналов звеньев согласно информации о способе их соединения для использования этих сигналов на следующем шаге аргумента.

Рассмотрим цифровые модели типовых линейных звеньев, полученные на основе метода Эйлера [3]. Индекс i в формулах обозначает номер итерации (дискретное время), а Δt – шаг дискретизации по времени.

1. Безынерционное звено (П-регулятор)

Передаточная функция $W(p) = K$.

Исходное уравнение $x_{\text{блх}}(t) = Kx_{\text{вх}}(t)$.

Цифровая модель $x_{i \text{ блх}} = Kx_{i \text{ вх}}$.

2. Идеальное интегрирующее звено (И-регулятор)

Передаточная функция $W(p) = \frac{K}{p}$.

Исходное уравнение $x'_{\text{блх}}(t) = Kx_{\text{вх}}(t)$.

Цифровая модель
$$\begin{cases} \Delta x_{i \text{ блх}} = x_{i \text{ вх}} \Delta t K; \\ x_{i \text{ блх}} = x_{i-1 \text{ блх}} + \Delta x_{i \text{ блх}}. \end{cases}$$

3. Аperiodическое звено 1-го порядка

Передаточная функция $W(p) = \frac{K}{Tp+1}$.

Исходное уравнение $Tx'_{\text{блх}} + x_{\text{блх}} = Kx_{\text{вх}}$.

$$\text{Цифровая модель} \begin{cases} x'_{i \text{ блх}1} = \frac{x_{i \text{ вх}} - x_{i-1 \text{ блх}1}}{T}; \\ x_{i \text{ блх}1} = x_{i-1 \text{ блх}1} + \Delta t x'_{i \text{ блх}1}; \\ x_{i \text{ блх}} = K x_{i \text{ блх}1}. \end{cases}$$

4. Изоморфное звено (ПИ-регулятор)

Передаточная функция $W(p) = \frac{1+T_0p}{T_1p}$.

Исходное уравнение $T_1x'_{\text{блх}} = x_{\text{вх}} + T_0x'_{\text{вх}}$.

$$\text{Цифровая модель} \begin{cases} \Delta x_{i \text{ блх}1} = \frac{x_{i \text{ вх}} \Delta t}{T_1}; \\ x_{i \text{ блх}1} = x_{i-1 \text{ блх}1} + \Delta x_{i \text{ блх}1}; \\ x_{i \text{ блх}2} = \frac{x_{i \text{ вх}} T_0}{T_1}; \\ x_{i \text{ блх}} = x_{i \text{ блх}1} + x_{i \text{ блх}2}. \end{cases}$$

5. Колебательное звено

Передаточная функция $W(p) = \frac{K}{T_0^2 p^2 + T_1 p + 1}$.

Исходное уравнение $T_0^2 x''_{\text{вых}} + T_1 x'_{\text{вых}} + x_{\text{вых}} = K x_{\text{вх}}$.

Цифровая модель
$$\begin{cases} x''_{i \text{ вых}1} = x_{i \text{ вх}} - T_1 x'_{i-1 \text{ вых}1} - x_{i-1 \text{ вых}1}; \\ \Delta x'_{i \text{ вых}1} = \frac{x''_{i \text{ вых}1} \Delta t}{T_0^2}; \\ x'_{i \text{ вых}1} = x'_{i-1 \text{ вых}1} + \Delta x'_{i \text{ вых}1}; \\ \Delta x_{i \text{ вых}1} = x'_{i \text{ вых}1} \Delta t; \\ x_{i \text{ вых}1} = x_{i-1 \text{ вых}1} + \Delta x_{i \text{ вых}1}; \\ x_{i \text{ вых}} = K x_{i \text{ вых}1}. \end{cases}$$

6. Консервативное звено

Передаточная функция $W(p) = \frac{K}{T_0^2 p^2 + 1}$.

Исходное уравнение $T_0^2 x''_{\text{вых}} + x_{\text{вых}} = K x_{\text{вх}}$.

Цифровая модель получается из предыдущего звена подстановкой $T_1 = 0$.

7. Интегрирующее звено с замедлением

Передаточная функция $W(p) = \frac{K}{p(1 + Tp)}$.

Исходное уравнение $Tx''_{\text{вх}} + x'_{\text{вх}} = Kx_{\text{вх}}$.

Цифровая модель
$$\begin{cases} x''_{i\text{ вх}1} = x_{i\text{ вх}} - x'_{i-1\text{ вх}1}; \\ \Delta x'_{i\text{ вх}1} = \frac{x''_{i\text{ вх}1} \Delta t}{T}; \\ x'_{i\text{ вх}1} = x'_{i-1\text{ вх}1} + \Delta x'_{i\text{ вх}1}; \\ \Delta x_{i\text{ вх}1} = x'_{i\text{ вх}1} \Delta t; \\ x_{i\text{ вх}1} = x_{i-1\text{ вх}1} + \Delta x_{i\text{ вх}1}; \\ x_{i\text{ вх}} = K x_{i\text{ вх}1}. \end{cases}$$

8. Идеальное дифференцирующее звено

Передаточная функция $W(p) = Kp$.

Цифровая модель $x_{\text{вх}} = Kx'_{\text{вх}}$.

Цифровая модель
$$\begin{cases} \Delta x_{i\text{ вх}} = x_{i\text{ вх}} - x_{i-1\text{ вх}}; \\ x_{i\text{ вх}} = \frac{\Delta x_{i\text{ вх}} K}{\Delta t}; \end{cases}$$

9. Дифференцирующее звено с замедлением

Передаточная функция $W(p) = \frac{T_1 p}{1 + T_0 p}$.

Исходное уравнение $T_0 x'_{\text{вх}} + x_{\text{вх}} = T_1 x'_{\text{вх}}$.

$$\text{Цифровая модель} \left\{ \begin{array}{l} x'_{i \text{ влх1}} = \frac{x_{i \text{ вх}} - x_{i-1 \text{ влх1}}}{T_0}; \\ x_{i \text{ влх1}} = x_{i-1 \text{ влх1}} + x'_{i \text{ влх1}} \Delta t; \\ x_{i \text{ влх}} = x'_{i \text{ влх1}} T_1. \end{array} \right.$$

10. Пропорционально-дифференциальный регулятор с замедлением

$$\text{Передающая функция } W(p) = \frac{K(T_1 p + 1)}{T_0 p + 1}.$$

$$\text{Исходное уравнение } T_0 x'_{\text{влх}} + x_{\text{влх}} = K(x_{\text{вх}} + T_1 x'_{\text{вх}}).$$

$$\text{Цифровая модель} \left\{ \begin{array}{l} x'_{i \text{ влх1}} = \frac{(x_{i \text{ вх}} - x_{i-1 \text{ влх1}})}{T_0}; \\ x_{i \text{ влх1}} = x_{i-1 \text{ влх1}} + x'_{i \text{ влх1}} \Delta t; \\ x_{i \text{ влх2}} = x'_{i \text{ влх1}} T_1; \\ x_{i \text{ влх3}} = x_{i \text{ влх1}} + x_{i \text{ влх2}}; \\ x_{i \text{ влх}} = K x_{i \text{ влх3}}. \end{array} \right.$$

Способ формализации структурных связей между отдельными звеньями системы рассмотрим на примере структурной схемы САУ, представляемой на рис. 1.1, а.

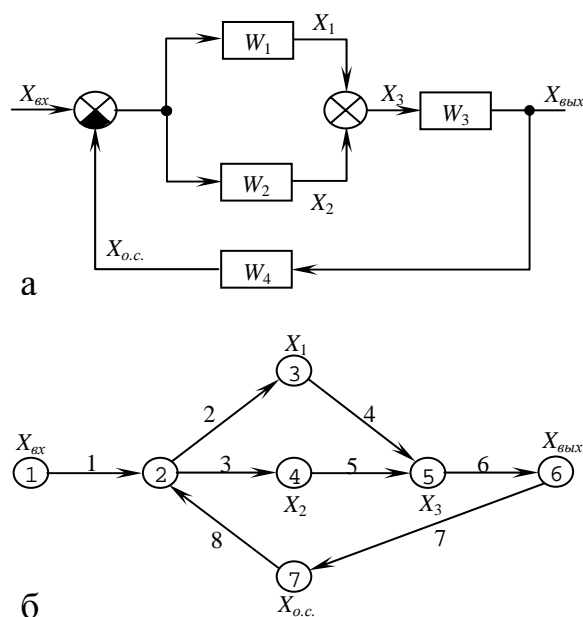


Рис. 1.1. Структурная схема САУ (а) и ее сигнальный граф (б)

Сигнальный граф этой системы изображен на рис. 1.1, б, в котором звенья структурной схемы представлены дугами, а сигналы – вершинами. Для полноты представления сигналов на графе введены единичные дуги с номерами: 1, 8, 4, 5. Топологические свойства (связи между вершинами графа) зададим одномерным массивом IS , элементы которого с четными индексами есть номера вершин, из которых выходит дуга, а элементы с нечетными индексами – номера вершин, в которые входят дуги [7]. Таким образом, каждая пара (соседних) элементов задает дугу графа в виде упорядоченной пары начальной и конечной вершин. Для рассматриваемого примера массив топологии графа будет содержать следующие элементы:

$$IS = \{2, 1, 3, 2, 4, 2, 5, 3, 5, 4, 6, 5, 7, 6, 2, 7\}.$$

На сигнальном графе (см. рис. 1.1, б) дуги пронумерованы в порядке появления их в массиве топологии IS таким образом, что IS_{2i} – начальная, а IS_{2i-1} – конечная вершины i -й дуги графа. Такой способ задания топологии системы является компактным и эффективным при моделировании САУ во временной области. Топологические связи между отдельными звеньями в

процессе моделирования могут быть реализованы с использованием следующего фрагмента программы, написанной на языке Си:

```
for (i = 1; i < N; i++) Xinput[i] = Xbegin[i];

/* начало цикла по времени */...

/* опрос звеньев на шаге аргумента t и вычисление Xoutput */...

for (j = 1; j < NG; j++)
{
Y[j] = 0.0;
for (i = 1; i < N; i++)
{
if (IS[2*i-1] == j)
Y[j] += Xoutput[i];
}
}

Y [MB] = input(t);
for (i = 1; i < N; i++)
{
k = IS[2*i];
Xinput [i]= Y[k];
}

... /* окончание цикла по времени */
```

В программе введены следующие обозначения: NG – число вершин графа; N – число дуг графа; Y[NG] – массив сигналов графа; Xinput[N], Xoutput[N] – соответственно входные и выходные сигналы звеньев; Xbegin[N] – начальные условия; MB – номер входной вершины; input()– функция для вычисления входного воздействия.

Кроме того, в программе должны быть предусмотрены переменные (или массив) для хранения переменных состояния звеньев, обладающих инерционностью. Важным средством обеспечения диалогового режима моделирования САУ является графический редактор для ввода структурной схемы и численных значений коэффициентов передаточных функций. Как следует из вышеизложенного, выходными данными редактора должны являться массив топологии IS, информация о типе и коэффициентах каждого звена.

В заключение следует отметить, что для повышения точности моделирования цифровые модели типовых звеньев, приведенные выше, можно строить на основе методов численного интегрирования более высоких порядков, например, метода Рунге-Куты четвертого порядка.

1.3. Структурное цифровое моделирование САУ в частотной области

Изменение комплексного коэффициента передачи системы в диапазоне частот ω от 0 до ∞ дает полную информацию о динамических свойствах системы управления. Геометрическое место точек конца вектора комплексного коэффициента передачи представляет собой комплексную частотную характеристику системы. Возможны два способа представления точек частотной характеристики. В прямоугольной системе координат:

$$(P(\omega), Q(\omega)),$$

$$\text{где } P(\omega) = \operatorname{Re}\{W(j\omega)\};$$

$$Q(\omega) = \operatorname{Im}\{W(j\omega)\}.$$

Другая форма представления точек частотной характеристики – в полярной системе координат:

$$(A(\omega), \varphi(\omega)),$$

$$\text{где } A(\omega) = |W(j\omega)| \text{ – модуль частотной характеристики;}$$

$$\varphi(\omega) = \arctg \left\{ \frac{\operatorname{Im}\{W(j\omega)\}}{\operatorname{Re}\{W(j\omega)\}} \right\} - \text{фаза (аргумент)}.$$

Как известно, получить выражение для комплексной частотной характеристики можно, заменив переменную Лапласа p на выражение $j\omega$ в формуле для передаточной функции системы. Таким образом, структурно-топологическая частотная модель системы управления может быть получена на основе ее структурной схемы. Используя операторную модель вида [4]

$$R_y = \langle W, C \rangle,$$

где C – бинарное отношение на множестве элементов системы, отражающее ее топологию; W – множество передаточных функций системы, и переходя от нее в частотную область, можно формализовать получение частотных характеристик как всей системы для любого выхода, так и отдельных конструктивно обособленных элементов системы. Формализация вычисления частотных характеристик основывается на машинных методах топологического анализа схем [7] для применения формулы Мезона [1]:

$$W(j\omega) = \sum_{i=1}^m [W_{pi}(j\omega) \Delta_i(j\omega)] / \Delta(j\omega), \quad (1.1)$$

где $W(j\omega)$ – комплексный коэффициент передачи всей системы; $W_{pi}(j\omega)$ – комплексный коэффициент передачи i -го пути от рассматриваемых входа до выхода системы; $\Delta(j\omega)$ – определитель графа системы;

$$\begin{aligned} \Delta(j\omega) = & 1 - \sum_i W_{0i}(j\omega) + \sum_{i,j} W_{0i}(j\omega) W_{0j}(j\omega) - \\ & - \sum_{i,j,k} W_{0i}(j\omega) W_{0j}(j\omega) W_{0k}(j\omega) + \dots, \end{aligned}$$

где $W_{0i}(j\omega)$ в первой сумме – передача i -го контура, равная произведению комплексных коэффициентов передач, входящих в этот контур дуг, и суммирование производится по всем контурам системы; $W_{0i}(j\omega) W_{0j}(j\omega)$ во второй сумме – произведение передач i -го и j -го контуров и суммирование производится по всем непересекающимся парам контуров; в третьей сумме – произведения непересекающихся троек контуров и т.д.; $\Delta_i(j\omega)$ – определитель подграфа, получающийся из исходного графа при удалении дуг и вершин i -го простого пути, а также всех дуг, инцидентных этим вершинам.

Два контура называются непересекающимися, если они не имеют общих дуг и (или) общих вершин. Тройка (четверка и т.д.) контуров называется непересекающейся, если любая пара контуров из этой тройки (четверки и т.д.) является непересекающейся.

Для анализа топологии САУ и реализации выражения (1.1) разработаны программные средства на различных языках программирования [7,8,9]. Представление топологической информации о системе в этих программных комплексах производится в виде бинарных матриц, определяющих составы путей, и контуров. Рассмотрим САУ, представленную на рис. 1.1, а. Граф этой системы имеет два контура, два пути от вершины $X_{вх}$ до вершины $X_{вых}$. Состав путей и контуров представим прямоугольными бинарными матрицами путей P и контуров C :

$$P = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \end{bmatrix};$$

$$C = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 \end{bmatrix},$$

где $p_{ij} = 1$ означает вхождение j -й дуги в i -й путь; $c_{ij} = 1$ – вхождение j -й дуги в i -й контур.

Отношение касания контуров между собой и путей и контуров представим бинарными матрицами Q и R соответственно:

$$Q = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}; \quad R = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix},$$

где $q_{ij} = 1$ означает касание i -го и j -го контуров, $r_{ij} = 1$ означает касание i -го пути и j -го контура.

Учитывая передачи дуг графа системы, запишем элементы формулы (1.1) через комплексные коэффициенты передачи звеньев:

$$W_{p1}(j\omega) = W_1(j\omega)W_3(j\omega);$$

$$W_{p2}(j\omega) = W_2(j\omega)W_3(j\omega);$$

$$\Delta(j\omega) = 1 + W_1(j\omega)W_3(j\omega)W_4(j\omega) + W_2(j\omega)W_3(j\omega)W_4(j\omega);$$

$$\Delta_1(j\omega) = 1;$$

$$\Delta_2(j\omega) = 1.$$

Подстановка этих выражений в формулу (1.1) дает общий комплексный коэффициент передачи системы.

1.4. Задание

1. Используя цифровые модели типовых динамических звеньев, приведенные выше, написать программу на языке высокого уровня для моделирования САУ во временной области (структурная схема САУ задаётся преподавателем).
2. Произвести моделирование САУ для различных типовых входных сигналов и параметров моделирования.
3. Оценить правильность и точность моделирования путем сравнения результатов с результатами моделирования, полученными при использовании других программных средств.

2. ПРИМЕНЕНИЕ СЕТЕЙ ПЕТРИ ДЛЯ МОДЕЛИРОВАНИЯ СИСТЕМ ЛОГИЧЕСКОГО УПРАВЛЕНИЯ

2.1. Особенности систем логического управления как объектов моделирования

Рассмотрим в качестве объекта моделирования систему логического управления (СЛУ) технологическим процессом. Функции технологического процесса разделяются на информационные и управляющие [12].

К информационным функциям относятся централизованный контроль за состоянием технологического объекта (измерение и регистрация значений технологических параметров, обнаружение и регистрация их отклонений, контроль срабатывания блокировок и защит и др.) и логико-вычислительные операции информационного характера (вычисление обобщенных показателей состояния оборудования, анализ срабатывания блокировок и защит, диагностика и т. д.).

К управляющим функциям относятся определение рационального режима ведения технологического процесса, формирование управляющих воздействий на объект, выдача оператору рекомендаций по управлению процессом. Укрупненная функциональная схема управления технологическим оборудованием приведена на рис. 2.1, где ТО – технологический объект; ИМ – исполнительные механизмы; Д – датчики; Р – средства регулирования; СЛУ – система логического управления; СВТ – средства вычислительной техники; В – временные средства (таймеры).

Проектирование системы управления предусматривает последовательное решение таких задач, как выбор ИМ и Д, комплектование средств регулирования, построение СЛУ. Для построения СЛУ вначале должен быть описан алгоритм его функционирования (алгоритм поведения), составляемый с учетом известных свойств управляемого объекта.

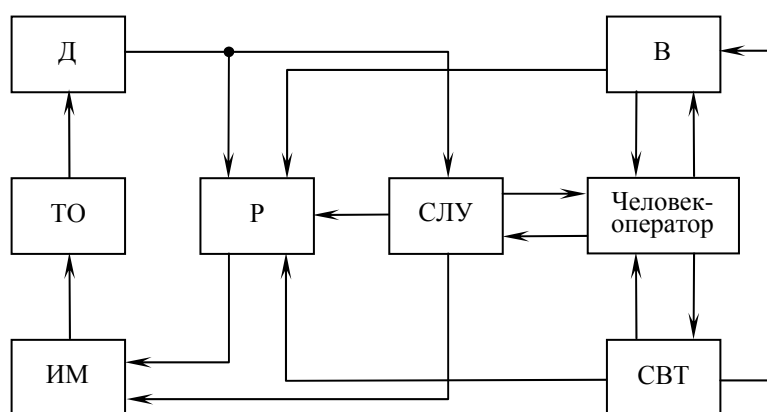


Рис. 2.1. Функциональная схема управления технологическим оборудованием

Применительно к СЛУ используется в основном структурная организация двух видов – асинхронная и с внутренней синхронизацией.

В СЛУ с асинхронной структурой любое возмущение происходит в результате изменения сигналов на его входе, образованном входными каналами элементов, которые не соединены с выходами других элементов. Возникающий при этом переходный процесс всегда завершается установлением равновесия, которое наступает прежде следующего изменения сигналов на входе, то есть переходный процесс происходит при неизменном состоянии входа. Система с асинхронной структурой должна функционировать таким образом, чтобы при любых значениях задержек в элементах по завершении любого переходного процесса равновесное состояние установилось однозначно, причем во время переходного процесса исключалась подача на выход непредусмотренных воздействий.

В СЛУ с внутренней синхронизацией осуществляется индикация реальных моментов окончания переходных процессов в элементах и использование этих сигналов индикации для синхронизации работы СЛУ с объектом управления в системе. Выбор формального языка, в наибольшей степени учитывающего особенности процесса управления, является основной задачей начального этапа проектирования СЛУ. Одно из характерных свойств

СЛУ, влияющих на структуру алгоритмов управления, заключается в наличии в ней множества блоков, часть которых может работать одновременно, взаимодействуя друг с другом. Поэтому формализованный язык должен обладать средствами для описания параллелизма. Широкое распространение многопрограммных и многопроцессорных систем вызвало развитие различных языковых средств, предназначенных для их описания и моделирования. Часть из них является расширением известных языков программирования, другие основаны на использовании различных графовых моделей. Не имея возможности дать полный обзор этих языков и провести их сравнение, рассмотрим более подробно язык сетей Петри [13-15], получивший широкое распространение и обладающий свойствами, которые позволяют в значительной степени учитывать структуру и характер функционирования СЛУ.

2.2. Основы теории сетей Петри

Появление теории сетей Петри связано с именем доктора Петри, защитившим в 1962 году диссертацию, в которой были разработаны основы теории моделирования дискретных процессов. Сети Петри разрабатывались специально для описания и моделирования систем, состоящих из отдельных взаимодействующих блоков. При этом допускается описание одновременно функционирующих блоков. Другой характерной особенностью является то, что каждый отдельный блок сам может быть целой системой и ее поведение возможно описать независимо от других блоков, за исключением точно определенного взаимодействия с этими блоками. Таким образом, сети Петри являются эффективным инструментом моделирования дискретно протекающих процессов с возможностью отображения таких свойств моделируемых объектов, как параллелизм, асинхронность, иерархичность.

Формализм теории сетей Петри основан на понятии комплекта, являющегося обобщением понятия множества. Комплект, как и множество, является набором элементов, но в отличие от него допускает кратное вхождение элементов. В случае множества элемент может входить или не входить во

множество, в случае комплекта элемент может входить в него целое, неотрицательное число раз. Комплектами, например, являются следующие наборы: $A_1 = \{a, b, b\}$; $A_2 = \{a, b, c\}$; $A_3 = \{b, b, c, c, d, d\}$.

Если взаимосвязь между элементами из некоторой области и множеством определяется функцией членства, принимающей значения 0 или 1, то взаимосвязь между элементами и комплектом определяется функцией числа вхождений, обозначаемой $\#(x, A)$ для элемента x и комплекта A . Так, для приведенных выше комплектов $\#(b, A_1) = 2$, $\#(a, A_2) = 1$, $\#(c, A_3) = 2$.

Из определения функции числа вхождений следует, что $\#(x, B) \geq 0$ для любых x и B , если $x \in B$, то $\#(x, B) > 0$, если $x \notin B$, то $\#(x, B) = 0$. Комплект является по определению пустым \emptyset , если для всякого x $\#(x, B) = 0$.

Мощностью $|A|$ комплекта A называется общее число вхождений элементов в комплект:

$$|A| = \sum_x \#(x, A).$$

Комплект A называется подкомплексом B ($A \subseteq B$), если всякий элемент A не меньшее число раз входит в B :

$$A \subseteq B \Leftrightarrow (\forall x) \#(x, A) \leq \#(x, B).$$

Над комплектами определены четыре операции:

1. Объединение комплектов A и B ($A \cup B$).

Объединением является комплект, для которого

$$\#(x, A \cup B) = \max (\#(x, A), \#(x, B)).$$

2. Пересечение комплектов A и B ($A \cap B$).

Функция числа вхождений для пересечения комплектов:

$$\#(x, A \cap B) = \min (\#(x, A), \#(x, B)).$$

3. Сумма комплектов A и B ($A + B$).

Функция числа вхождений для суммы комплектов:

$$\#(x, A + B) = \#(x, A) + \#(x, B).$$

4. Разность комплектов A и B ($A - B$).

Функция числа вхождений для результирующего комплекта

$$\#(x, A - B) = \#(x, A) - \#(x, B).$$

Пусть D – множество элементов, из которых строятся комплекты. Пространством комплектов D^n называется множество комплектов, элементы которых принадлежат D и никакой элемент не входит в комплект более n раз. Иначе говоря, для всех $B \in D^n$ из $x \in B$ следует $x \in D$ и $\#(x, B) \leq n$ для любого x . D^∞ – это множество всех комплектов над D .

Сеть Петри C – это четверка $C = \langle P, T, I, O \rangle$, где $P = \{p_1, p_2, \dots, p_n\}$ – множество позиций ($n > 0$); $T = \{t_1, t_2, \dots, t_m\}$ – множество переходов ($m \geq 0$), $P \cap T = \emptyset$, I – входная функция-отображение переходов в комплекты позиций:

$$I : T \rightarrow P^\infty;$$

O – выходная функция-отображение переходов в комплекты позиций:

$$O : T \rightarrow P^\infty.$$

Если $p_i \in I(t_j)$, то позиция p_i называется входной для перехода t_j , если $p_i \in O(t_j)$, то позиция p_i называется выходной для перехода t_j . Кратность входной позиции p_i для перехода t_j – это число вхождений позиции во входной комплект, т.е. $\#(p_i, I(t_j))$ аналогично кратность выходной позиции p_i для перехода t_j – это число вхождений позиции в выходной комплект перехода, т.е. $\#(p_i, O(t_j))$. Для иллюстрации понятий теории сетей Петри гораздо более удобно графическое представление. Графическим представлением сети Петри является двудольный граф, множество вершин которого образуется объединением множеств $P \cup T$, а смежность вершин задается функциями I и O . Позиции на графе изображаются кружками, переходы – палочками. Дуга соединяет позицию и переход (направлена от позиции к переходу), если позиция является входной для перехода. Дуга направлена от перехода к позиции, если позиция является выходной для перехода. Кратные входы и выходы

изображаются кратными дугами, поэтому сеть Петри в общем случае является ориентированным двудольным мультиграфом. Рассмотрим пример сети Петри $C = \langle P, T, I, O \rangle$.

$$\begin{aligned} I(t_1) &= \{p_1\}, & O(t_1) &= \{p_1, p_2, p_3\}, \\ I(t_2) &= \{p_2, p_3, p_5\}, & O(t_2) &= \{p_5\}, \\ I(t_3) &= \{p_3\}, & O(t_3) &= \{p_4, p_4\}, \\ I(t_4) &= \{p_4\}. & O(t_4) &= \{p_2, p_3\}. \end{aligned}$$

Граф этой сети изображен на рис. 2. 2

Маркировка μ сетей Петри $C = \langle P, T, I, O \rangle$ – это функция, отображающая множество позиций P в множество неотрицательных целых чисел N , $\mu : P \rightarrow N$. Другими словами, маркировка присваивает каждой позиции некоторое, быть может, нулевое, число меток. На графе метки отображаются точками внутри позиций. Сеть Петри с определенной на ней разметкой называется маркированной. Маркировку μ можно определить как n -мерный вектор $\mu = (\mu_1, \dots, \mu_n)$, где $n = |P|$ – число позиций, а $\mu_i \in N$ – число меток в позиции p_i .

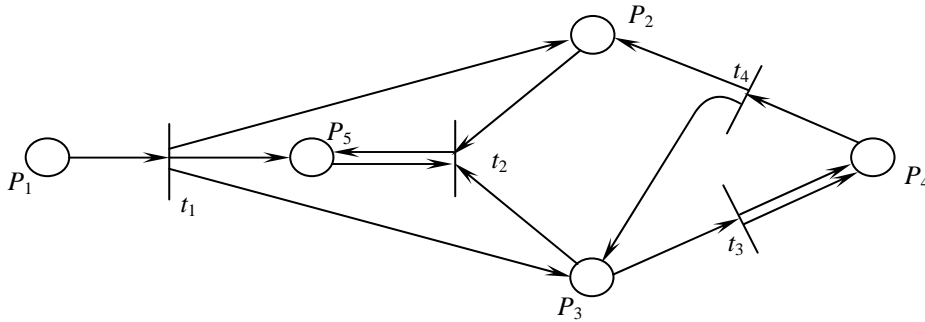


Рис. 2. 2. Пример графа сети Петри

Число и расположение меток могут изменяться при выполнении сети Петри, которое само зависит от числа и распределения меток по сети. Под выполнением сети Петри понимается последовательность запусков переходов. Переход запускается удалением меток из его входных позиций и добав-

лением меток в выходные позиции перехода. Переход может запускаться только в том случае, если он разрешен. Переход называется разрешенным, если каждая из его входных позиций имеет число меток, по крайней мере, равное числу дуг из позиции в переход. Метки во входных позициях, которые разрешают переход, называются его разрешающими метками.

Формальное определение: переход t_j в сети с маркировкой μ разрешен, если для всех $p_i \in P$ $\mu(p_i) \geq \#(p_i, I(t_j))$, где $\mu(p_i)$ – количество меток в позиции p_i .

Переход запускается удалением всех его разрешающих меток из входных позиций и последующим помещением в каждую из его выходных позиций по одной метке для каждой выходящей из перехода дуги. В общем случае запуск перехода заменяет маркировку сети Петри μ на новую маркировку μ' . Запуск перехода t_j приводит к новой маркировке μ' , определяемой из соотношения:

$$\mu'(p_i) = \mu(p_i) - \#(p_i, I(t_j)) + \#(p_i, O(t_j)).$$

Поскольку запускаться могут только разрешенные переходы, для которых $\mu(p_i) \geq \#(p_i, I(t_j))$ для всех p_i , то число меток в любой позиции будет оставаться неотрицательным.

Рассмотрим маркированную сеть на рис. 2.3, а. В этой сети переход t_1 разрешен, а переход t_2 запрещен, так как отсутствует разрешающая метка в позиции p_3 . После срабатывания перехода t_1 разметка изменится, как показано на рис. 2.3, б. В этом состоянии разрешен переход t_2 , а переход t_1 запрещен, так как отсутствует разрешающая метка в позиции p_1 . С выполнением сети Петри связаны две последовательности: последовательность маркировок $(\mu_0, \mu_1, \mu_2, \dots)$ и последовательность переходов $(t_{j_0}, t_{j_1}, t_{j_2}, \dots)$. Эти две последовательности связаны соотношением

$$\delta(\mu_k, t_{j_k}) = \mu_{k+1}, \quad k = 0, 1, 2, \dots$$

Имея последовательность переходов и μ_0 , легко получить последовательность маркировок, но не наоборот.

Пусть некоторый переход в маркировке μ разрешен и, следовательно, может быть запущен. Результат запуска перехода в маркировке μ есть новая маркировка μ' . Говорят, что μ' является непосредственно достижимой из маркировки μ .

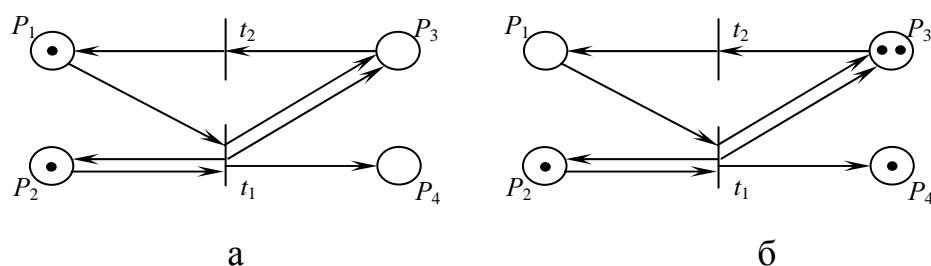


Рис. 2.3. Маркированная сеть Петри: а – до срабатывания перехода t_1 ;
б – после срабатывания перехода t_1

2.3. Моделирование с помощью сетей Петри

Простое представление системы сетью Петри основано на двух основополагающих понятиях: событиях и условиях. События – это действия, имеющие место в системе. Возникновением событий управляет состояние системы. Состояние системы может быть описано множеством условий. Условие – это предикат или логическое описание состояния системы. Условие может принимать либо значение «истина», либо значение «ложь».

Условия, выполнение которых необходимо для возникновения события, называются его предусловиями. При возникновении события система переходит в состояние, характеризующееся выполнением условий, называемых постусловиями.

Рассмотрим задачу моделирования логического управления технологическим участком, на котором детали проходят весь путь обработки.

Для этой системы выделим следующие условия:

А – деталь ожидает обработки;

Б – технологический робот свободен;

В – деталь обрабатывается;

Г – деталь ожидает транспортного робота.

Событиями в системе являются следующие действия:

1 – деталь поступает во входную очередь;

2 – деталь начинает обрабатываться;

3 – обработка детали завершена;

4 – передача детали в накопитель.

Связь событий с предусловиями и постусловиями показана в табл. 2.1.

Таблица 2.1

Событие	Пред- условие	Пост- условие
1	Нет	А
2	А, Б	В
3	В	Г, Б
4	Г	Нет

Эта таблица интерпретируется сетью Петри следующим образом. Условия представляются позициями, события – переходами. Входами перехода являются предусловия соответствующего события, выходами – его постусловия. Возникновение события моделируется запуском перехода. Выполнение условия моделируется наличием метки в позиции, соответствующей условию. Сеть, моделирующая описанную систему, приведена на рис. 2.4.

Выполнение сети Петри (и соответственно поведение моделируемой системы) рассматривается как последовательность дискретных событий.

Последовательность возникновения событий может быть реализована любая из множества возможных. Если в какой-либо момент времени разрешено несколько переходов, то следующим запускаемым переходом может

быть любой из них. Это свойство сетей Петри отражает неоднозначность порядка возникновения событий, связанных с параллельно протекающими процессами. Таким образом, выполнению сети Петри свойственен недетерминизм.

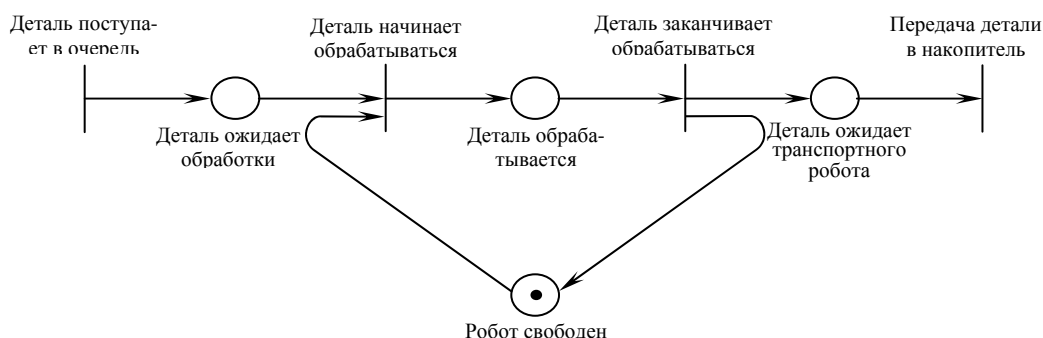


Рис. 2.4. Модель технологического участка в виде сети Петри

Другая важная особенность сетей Петри – их асинхронная природа. Сети Петри не обладают какими-либо средствами, отражающими течение времени или фиксирующими некоторые моменты времени. Однако они представляют возможность отображать частичный порядок возникновения событий, т.е. обладают важным свойством отражения относительного времени протекания процессов путем установления причинно-следственной связи событий.

Средства установления причинно-следственных связей событий позволяют моделировать важные свойства систем логического управления – параллелизм и конфликтные ситуации. Параллелизм моделируется в сети Петри независимыми разрешенными переходами (рис. 2.5, а). Переходы t_j и t_k в такой ситуации не влияют друг на друга, поэтому возможные последовательности срабатывания переходов здесь такие $\dots t_j, \dots, t_k, \dots$ и $\dots t_k, \dots, t_j, \dots$

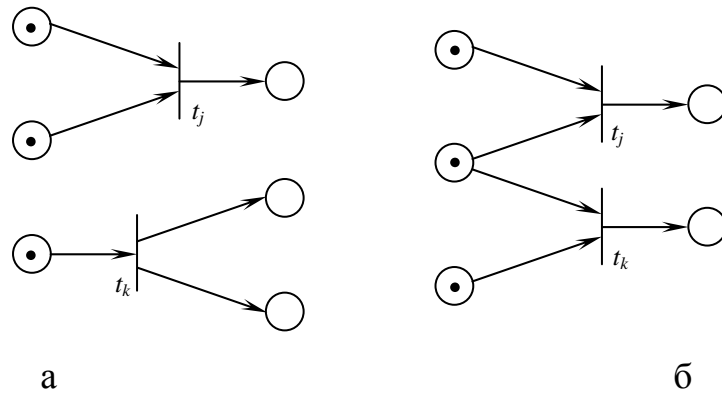


Рис. 2.5. Представление в терминах сетей Петри: а – параллельных процессов; б – конфликтных ситуаций

В другой ситуации (рис. 2.5, б), называемой конфликтной, разрешенные переходы имеют общую разрешающую метку, и поэтому запуск одного перехода удаляет общую метку и таким образом запрещает другой переход.

Таким образом, сети Петри – удобный инструмент для моделирования систем, события в которых происходят асинхронно и независимо, и представления причинно-следственных связей, на основе чего моделируются параллелизм и конфликты.

Введение параллелизма полезно только в том случае, когда компоненты процессов могут взаимодействовать при решении задачи. Такое взаимодействие требует распределения ресурсов между процессами. Для обеспечения правильности работы системы в целом распределением необходимо управлять. При взаимодействии процессов возникает проблема синхронизации. Рассмотрим моделирование некоторых разновидностей задачи синхронизации.

1. Задача о производителе-потребителе

Пусть процесс-производитель создает объекты, которые помещаются в буфер. Потребитель ждет, пока объект не будет помещен в буфер, удаляет его оттуда и использует. Такая ситуация моделируется сетью, представленной на рис. 2.6. Позиция B представляет собой буфер, каждая метка соответствует элементу, который произведен, но еще не использован.

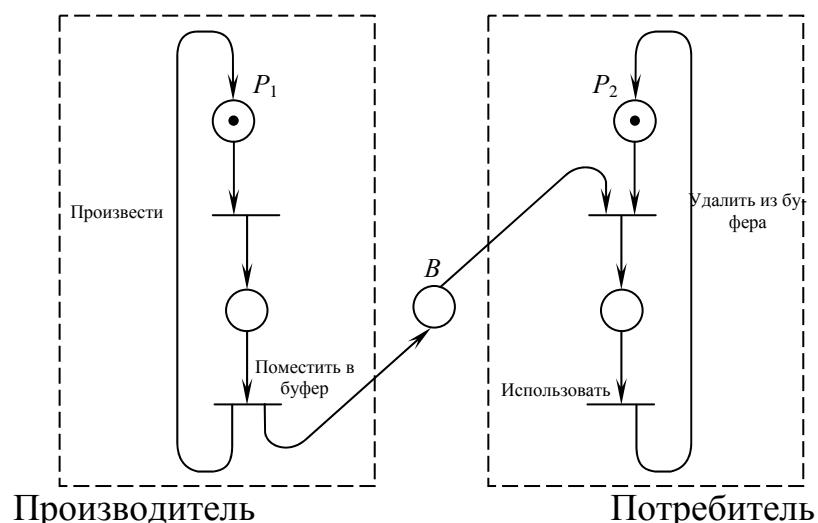


Рис. 2.6. Сетевая модель задачи о производителе-потребителе

Разновидность этой задачи – задача о нескольких производителях-потребителях. Если имеем t потребителей и s производителей, начальная маркировка сети изменится. В позиции p_1 будет s меток, в позиции p_2 – t меток. Третий вариант задачи – задача о производителе-потребителе с ограниченным буфером, т.е. буфер имеет только n ячеек элементов данных. Следовательно, производитель не имеет возможности работать с постоянной скоростью и иногда вынужден ждать, если буфер заполнен, т.е. потребитель работает медленно.

Сеть, моделирующая эту ситуацию, представлена на рис. 2.7.

Ограниченному буферу соответствует две позиции: B и B' . Разметка в B соответствует количеству помещенных в буфер элементов данных, разметка в B' – количеству свободных мест хранения в буфере. Начальная разметка соответствует показанной на графе (n меток в позиции B').

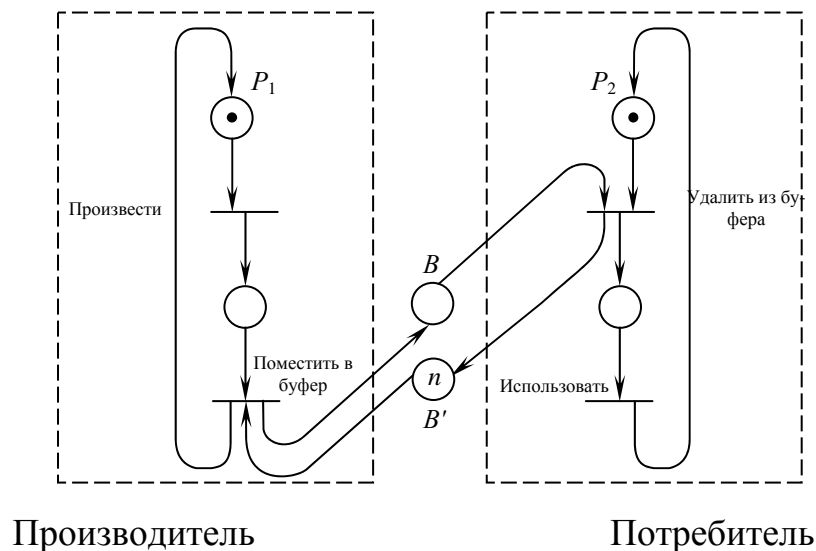


Рис. 2.7. Сетевая модель задачи о производителе-потребителе с ограниченным буфером

Отсутствие меток в позиции B' соответствует полностью заполненному буферу. В этом случае переход «поместить в буфер» запрещен, и производитель будет ожидать события «удалить из буфера», которое приведет к появлению разрешающей метки в B' .

2. Задача о чтении-записи

Пусть имеются процессы двух типов: процессы чтения и процессы записи. Все процессы совместно используют общий файл или элемент данных. Процессы чтения не изменяют этот общий объект в отличие от процессов записи. Таким образом, каждый процесс записи должен исключать все другие процессы чтения и записи, в то время как несколько процессов чтения могут иметь доступ к разделяемым данным одновременно. Задача состоит в определении структуры управления, которая не приведет к тупику и не допустит нарушения критерия взаимного исключения. На рис. 2.8 иллюстрируется решение задачи, когда количество процессов чтения ограничено числом s ($s < n$).

Двойные стрелки указывают кратные дуги (n -кратность дуг). Первоначально имеется s процессов чтения и t процессов записи. Инициализация процесса чтения моделируется запуском перехода t_1 (удаление по одной метке из позиции p_1 и p_2). Тем самым исключается запуск перехода t_2 (инициализация процесса запуска), поскольку число разрешающих меток в позиции p_2 становится меньше кратности n дуг. Аналогично могут быть инициализированы все остальные $s-1$ процессов чтения и лишь по завершении их всех (при наличии n меток в позиции p_2) возможен запуск перехода t_2 , т.е. инициализация записи.

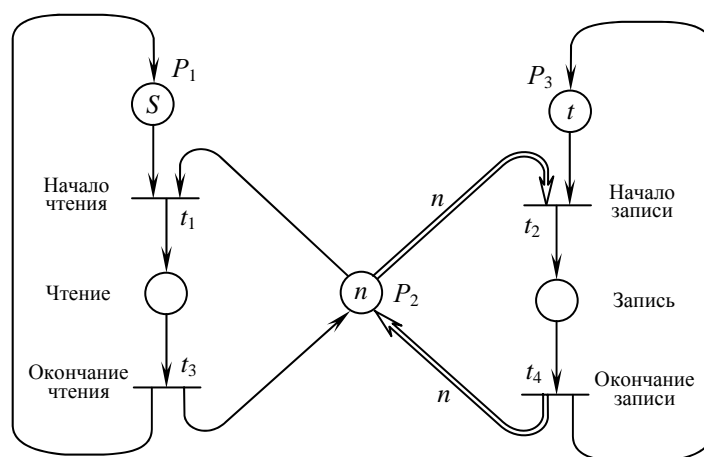


Рис. 2.8. Модель задачи о чтении-записи

3. Моделирование двухстаночной системы с роботом манипулятором

На рис. 2.9 приведена модель двухстаночной системы, обслуживаемой роботом. Робот представлен динамическим элементом-меткой. Ее начальная позиция p_1 соответствует обслуживанию станка 1, а позиция p_2 – обслуживанию станка 2.

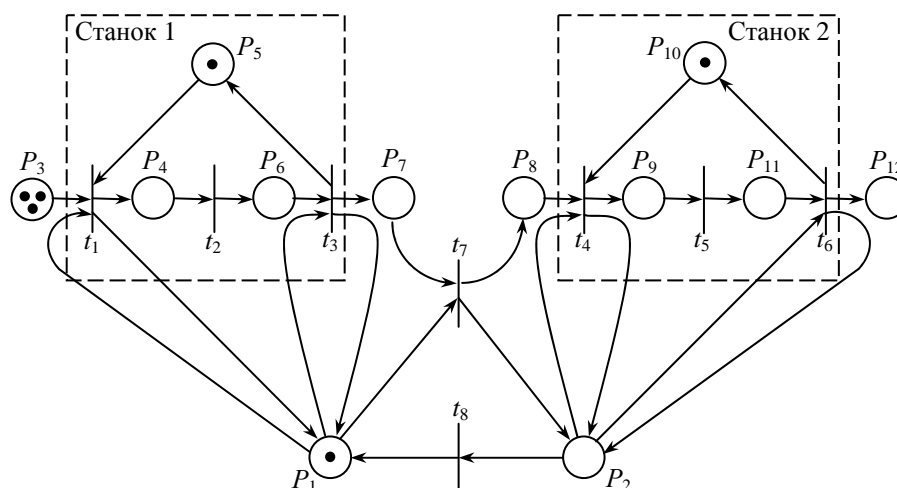


Рис. 2.9. Модель двухстаночной системы, обслуживаемой роботом-манипулятором

Партия обрабатываемых деталей помещается перед началом работы в бункер, моделируемый позицией p_3 . Обработанные на первом станке детали помещаются в накопитель (позиция p_7). Переходы t_1 , t_2 и t_3 представляют соответственно операции загрузки, окончания обработки и разгрузки станка. Позиция p_5 и ее управляющая метка создают условия для выполнения этих операций только с одной деталью. Переход метки в позицию p_4 соответствует окончанию загрузки станка, а в p_6 – завершению обработки. Модель второго станка имеет аналогичную структуру, представляя соответствующие фазы обслуживания. Переход t_7 моделирует транспортировку детали роботом от станка 1 к станку 2, а t_8 – имитирует обратный ход робота от станка 2 к станку 1.

4. Задача о распределении ресурсов

С этой задачей тесно связана задача предотвращения тупиков. Рассмотрим задачу предотвращения тупика на следующем примере (рис. 2.10).

Процессы A и B нуждаются в ресурсах $q(p_4)$ и $r(p_5)$. Начальная маркировка показывает, что процессы готовы, а ресурсы доступны (метки в p_1, p_6, p_4, p_5). Если будут запущены последовательно переходы t_1 и t_4 , то возникает ситуация взаимного блокирования процессов (состояние тупика) в результате того, что процесс A будет обладать ресурсом $q(p_4)$ и ждать ресурс $r(p_5)$, а процесс B – наоборот.

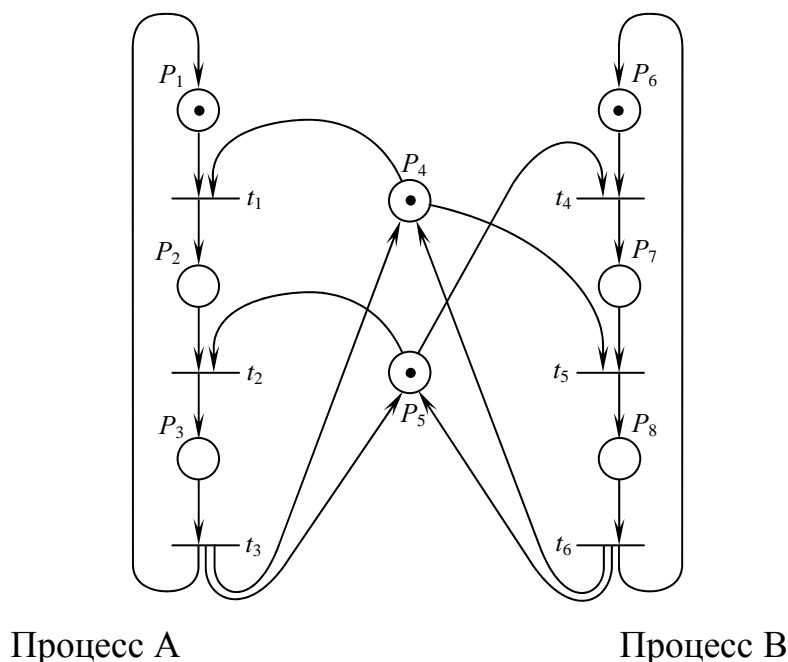


Рис. 2.10. Модель задачи о распределении ресурсов

Аналогичная ситуация возникает при последовательном запуске сначала t_4 , затем t_1 . Для предотвращения тупиковых ситуаций необходимо внести изменения в структуру модели, а значит и в алгоритм функционирования системы. Например, введение дополнительных дуг – от p_6 к t_1 , от p_1 к t_4 , от t_3 к p_6 , от t_6 к p_1 – при начальной разметке, приведенной на графе, устраняет появление тупиков, но при этом процессы A и B инициализируются поочередно.

2.4. Свойства сетей Петри

Рассмотрим свойства сетей Петри, наличие которых желательно с некоторых точек зрения для моделей систем логического управления. Метки в

позициях означают либо выполнение, какого-либо условия, например передачу управления некоторому блоку, либо непосредственно соответствуют некоторым материальным объектам или информационным единицам. Это накладывает определенные требования на максимально возможное количество меток в позициях сети. Подобные требования накладываются на сеть Петри и при аппаратурной ее реализации. Позиция сети может реализовываться счетчиком, состояние которого говорит о количестве меток в позиции. Поэтому для адекватной работы такой аппаратурно-реализованной сети Петри необходимо, чтобы число меток в позиции не превышало максимально представимого счетчиком числа. Эти соображения приводят к понятию ограниченности сети.

Позиция p_i сети Петри $C = \langle P, T, I, O \rangle$ с начальной маркировкой μ называется n -ограниченной, если число меток в ней не может превышать целого числа n , или

$$\mu'(p_i) \leq n, \quad \forall \mu' \in R(c, \mu),$$

где $R(c, \mu)$ – множество всех достижимых из начальной маркировки маркировок (множество достижимости).

Каждая позиция может иметь свое значение n , если такое n вообще существует. Сеть Петри ограничена, если ограничены все ее позиции. Если n – максимальное число меток, которое может быть в одной позиции ограниченной сети Петри, то сеть называют n ограниченной. Ограниченную сеть можно реализовать аппаратно. Особый интерес представляют один ограниченные позиции и сети, которые получили название безопасных. Безопасные позиции можно реализовать аппаратно, используя для этого триггеры. Кроме того, безопасные позиции можно интерпретировать как логические условия: при наличии в них метки условие выполняется, при отсутствии – нет.

Как было показано выше, сети Петри можно использовать для моделирования систем распределения ресурсов. Например, сеть может моделиро-

вать запросы, распределения и освобождения устройств ввода-вывода в вычислительной системе. В этих моделях систем некоторые метки могут составлять ресурсы. Для сетей такого типа в результате запуска переходов метки, представляющие ресурсы, не уничтожаются и не создаются, т.е. наблюдается своего рода сохранение меток.

Сеть Петри $C = \langle P, T, I, O \rangle$ с начальной маркировкой μ называется строго сохраняющей, если

$$\sum_{p_i \in P} \mu'(p_i) = \sum_{p_i \in P} \mu(p_i), \quad \forall \mu' \in R(c, \mu),$$

т.е. общее количество меток в процессе выполнения сети остается постоянным. Строгое сохранение – это очень сильное ограничение. Из него немедленно следует, что число входов в каждый переход должно равняться числу выходов, т.е.

$$|I(t_j)| = |O(t_j)|, \quad \forall j.$$

Если бы это было не так, то запуск перехода изменил бы число меток в сети.

На рис. 2.11 приведен пример сети, не являющейся строго сохраняющей, поскольку запуск переходов t_1 или t_2 уменьшает число меток на единицу, а запуски переходов t_3 или t_4 увеличивают это количество на единицу. Сеть является два ограниченной, поскольку позиции p_3 и p_4 два ограничены, а остальные позиции один ограничены (безопасны).

Сеть Петри с начальной маркировкой μ называется сохраняющей по отношению к вектору взвешивания $\omega = (\omega_1, \omega_2, \dots, \omega_n)$, если

$$\sum_i \omega_i \mu'(p_i) = \sum_i \omega_i \mu(p_i), \quad \forall \mu' \in R(c, \mu).$$

Строго сохраняющая сеть является сохраняющей по отношению к вектору взвешивания $\omega = (1, 1, \dots, 1)$. Так рассматриваемая сеть (рис. 2.11) является сохраняющей по отношению к вектору взвешивания $\omega = (1, 1, 1, 1, 0)$.

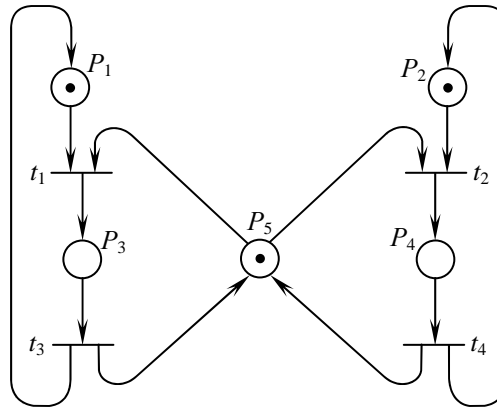


Рис. 2.11. Пример сети, иллюстрирующий свойства ограниченности и сохранения

Рассмотренные свойства сетей Петри образуют основу подхода к моделированию путем построения модели по системе, анализа модели с целью определения характеристик системы и модификации системы, если характеристики неудовлетворительны.

2.5. Методы анализа сетей Петри

Изложим методы анализа сетей Петри, позволяющие определить наличие или отсутствие в них рассмотренных выше свойств. Рассмотрим два основных метода анализа:

1. Построение дерева достижимости.
2. Метод матричных уравнений.

2.5.1. Построение дерева достижимости

Рассмотрим маркированную сеть, представленную на рис.2.12. Её начальная маркировка $\mu_0 = (1, 0, 0)$.

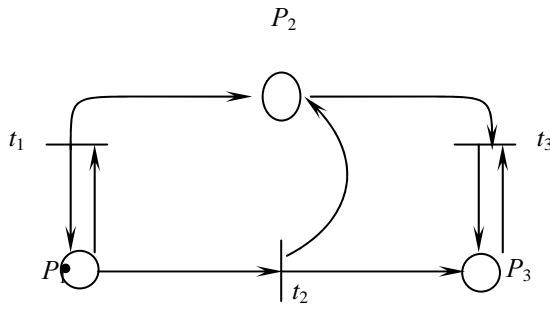
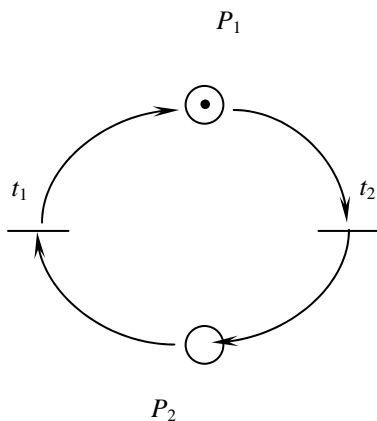
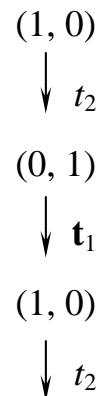


Рис. 2.12. Пример маркированной сети для построения дерева достижимости

В этой начальной маркировке разрешены два перехода t_1 , t_2 . Поскольку мы хотим рассмотреть всё множество достижимости, определим новые вершины в дереве достижимости для достижимых маркировок, получающихся в результате запуска каждого из этих двух переходов. Дерево достижимости – это исходящее дерево, вершинам которого соответствуют достижимые маркировки. Начальной маркировке соответствует корневая вершина. Если из вершины μ' в вершину μ'' ведет дуга, то она взвешена переходом, переводящим сеть Петри из маркировки μ' в μ'' . Тупиковым маркировкам соответствуют висячие вершины. Необходимо отметить, что в таком определении дерево достижимости в общем случае является бесконечным, поскольку даже сеть с конечным множеством достижимости (рис. 2.13, а) может иметь бесконечное дерево достижимости (рис. 2.13, б).



а



б

Рис. 2.13. Сеть с конечным множеством достижимости (а) и бесконечным деревом достижимости (б)

Для анализируемой сети (см. рис. 2.12) дерево достижимости представлено на рис. 2.14.

Всякий путь от корня в дереве соответствует допустимой последовательности переходов. Для превращения дерева в полезный инструмент анализа необходимо найти средства ограничения его до конечного размера, т.е. найти средства, которые ограничивают введение новых маркировок на каждом шаге построения. В этом смысле выделяют пассивные (терминальные), дублирующие вершины и вершины с расширенной маркировкой.

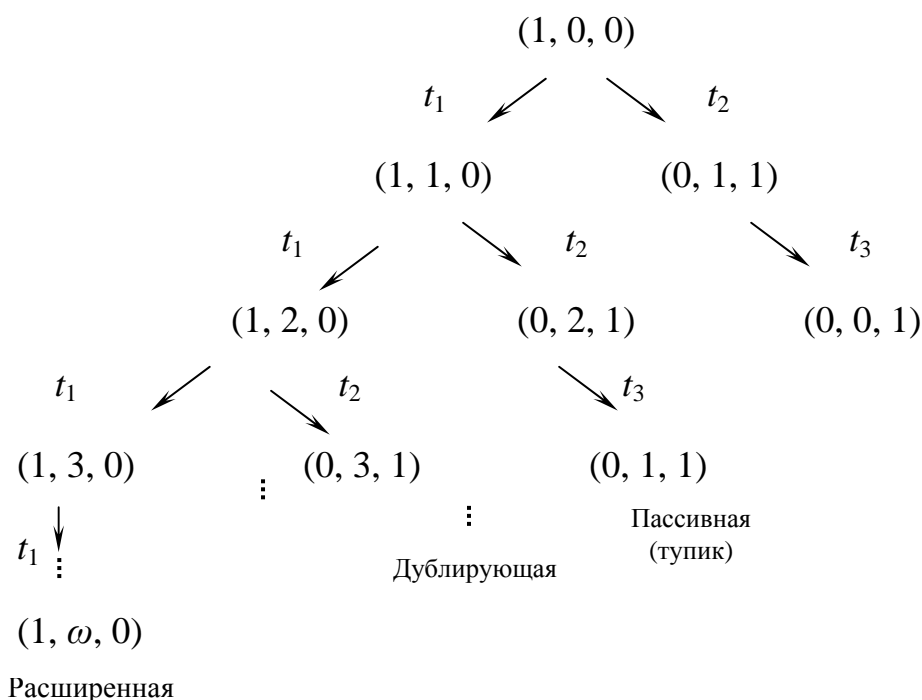


Рис. 2.14. Дерево достижимости

Пассивная (или терминальная) вершина – это маркировка, в которой нет разрешенных переходов. Для анализируемого примера сети это маркировка $(0, 0, 1)$ (см. рис. 2.12). Дублирующая вершина – это вершина с маркировкой, которая уже ранее встречалась в этом дереве достижимости. При появлении дублирующей вершины нет необходимости производить из нее дальнейшие построения, поскольку это приведет к появлению поддерева, полученного из места первого появления дублирующей вершины. В рассматриваемом примере маркировка $(0, 1, 1)$, получившаяся после выполнения t_1 , t_2 , t_3 не будет порождать какие-либо новые вершины, поскольку она

ранее была получена в дереве после выполнения t_2 из начальной маркировки.

Расширенная маркировка вводится, исходя из следующих соображений. Пусть существует последовательность запуска переходов σ , переводящая маркировку μ в μ' , причем маркировка μ' совпадает с μ за исключением того, что имеет дополнительные метки в некоторых позициях, т.е.

$$\mu' = \mu + (\mu' - \mu) \quad \text{и} \quad \mu' - \mu > 0.$$

Поскольку на запуск переходов лишние метки повлиять не могут, то последовательность σ можно запустить еще раз из μ' и получить μ'' , причем

$$\mu'' = \mu' + (\mu' - \mu) = \mu + 2(\mu' - \mu).$$

В общем случае можно запустить последовательность переходов σ n раз и получить

$$\mu''' = \mu' + n(\mu' - \mu).$$

Поскольку число n можно выбрать сколь угодно большим, в некоторых позициях сети разметка будет неограниченно возрастать. При построении дерева достижимости этот фрагмент дерева можно ограничить, вводя символ ω , который означает «бесконечность» в соответствующей позиции разметки. Для рассматриваемого примера (см. рис. 2.14) последовательное срабатывание t_1 приводит к нарастанию разметки в позиции p_2 , что приводит к введению расширенной маркировки $(1, \omega, 0)$.

Кроме того, различают граничные вершины и внутренние. Граничными вершинами называют вершины дерева достижимости, которые еще не обработаны алгоритмом построения дерева достижимости. В результате реализации алгоритма построения дерева достижимости граничные вершины превращаются в терминальные, дублирующие или внутренние.

Наличие свойств безопасности и ограниченности легко проверяется из дерева достижимости. Сеть Петри ограничена тогда и только тогда, когда символ ω отсутствует в дереве достижимости. Если символа ω нет, то можно определить границу числа меток в позициях путем перебора вершин де-

рева достижимости. Если в дереве достижимости присутствует символ ω , то для того, чтобы сеть была сохраняющей по отношению к некоторому вектору взвешивания, необходимо, чтобы вес ω_i позиции p_i , имеющей в некоторой маркировке ω , был нулевым.

2.5.2. Метод матричных уравнений

Этот подход основан на матричном представлении сетей Петри. Входная и выходная функции представляются матрицами D^- и D^+ соответственно. Элементы матриц определяются следующим образом:

$$D^- [i, j] = \# (p_i, I(t_j));$$

$$D^+ [i, j] = \# (p_i, O(t_j));$$

$$i = \overline{1, n}; \quad j = \overline{1, m}.$$

Матричная форма представления сети $C = \langle P, T, D^-, D^+ \rangle$ эквивалентна стандартной, но позволяет дать определения в терминах векторов и матриц.

Пусть $e[j]$ – m -вектор-столбец, содержащий все нули, за исключением j -й позиции. Переход t_j , таким образом, может быть представлен вектором $e[j]$.

Переход t_j в маркировке μ разрешен, если $\mu \geq D^- e[j]$, а результат запуска перехода t_j в маркировке μ равен

$$\sigma(\mu, t_j) = \mu' = \mu - D^- e[j] + D^+ e[j] = \mu + (D^+ - D^-) \cdot e[j],$$

где $(D^+ - D^-) = D$ – составная матрица изменений, элементы которой могут быть отрицательными.

Тогда для последовательности запусков переходов $\sigma = t_{j_1}, t_{j_2}, \dots, t_{j_k}$ имеем

$$\begin{aligned} \delta(\mu, \sigma) &= \mu + De[j_1] + De[j_2] + \dots + De[j_k] = \\ &= \mu + D(e[j_1] + e[j_2] + \dots + e[j_k]) = \mu + D \cdot f(\sigma), \end{aligned}$$

где $f(\sigma) = (e[j_1] + e[j_2] + \dots + e[j_k])$ – вектор запусков последовательности σ , каждый элемент которого показывает число запусков соответствующего пе-

рехода. Все элементы этого вектора неотрицательны.

Для того чтобы показать сохранение сети, необходимо найти (ненулевой) вектор взвешивания ω , размерностью $(1 \times n)$, для которого взвешенная сумма по всем маркировкам постоянна, т.е. $\omega\mu = \omega\mu'$.

Поскольку маркировка μ' достижима, существует последовательность запусков переходов σ , которая переводит сеть из μ в μ' :

$$\mu' = \mu + Df(\sigma).$$

Следовательно,

$$\omega\mu = \omega\mu' = \omega(\mu + Df(\sigma)) = \omega\mu + \omega Df(\sigma).$$

Отсюда следует, что $\omega Df(\sigma) = 0$. Поскольку это должно быть справедливо для всех $f(\sigma)$, то $\omega D = 0$.

Таким образом, сеть Петри является сохраняющей тогда и только тогда, когда существует такой вектор ω , что $\omega D = 0$. Если $\omega = (1, 1, \dots, 1)$, то это условие формулируется следующим образом: сумма элементов каждого столбца матрицы D должна быть нулевой.

Матричная теория сетей Петри является инструментом для решения проблемы достижимости. Предположим, что маркировка μ' достижима из маркировки μ . Тогда существует последовательность запусков переходов σ , которая приводит из μ к μ' . Это означает, что $f(\sigma)$ является неотрицательным целым решением следующего матричного уравнения для x :

$$\mu' = \mu + Dx \quad . \quad (2.1)$$

Если μ' достижима из μ , тогда уравнение имеет решение в неотрицательных целых, если уравнение не имеет решения, то μ' недостижима из μ . Наличие решения уравнения (2.1) является необходимым, но не достаточным для достижимости. Необходимо проверить, существует ли разрешенная последовательность запуска переходов, соответствующая вектору разметки μ' .

Рассмотрим сеть, представленную на рис. 2.15.

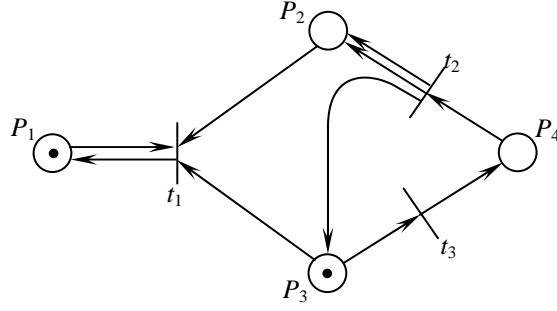


Рис. 2.15. Пример для иллюстрации матричного представления сетей Петри

Матрицы D^- , D^+ , D для этой сети имеют следующий вид:

$$D^- = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}; \quad D^+ = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}; \quad D = \begin{bmatrix} 0 & 0 & 0 \\ -1 & 2 & 0 \\ -1 & 1 & -1 \\ 0 & -1 & 1 \end{bmatrix}.$$

В начальной маркировке $\mu = (1, 0, 1, 0)^T$ переход t_3 разрешен и приводит к маркировке:

$$\mu' = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ -1 & 2 & 0 \\ -1 & 1 & -1 \\ 0 & -1 & 1 \end{bmatrix} \times \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ -1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix}.$$

Последовательность $\sigma = t_3, t_2, t_3, t_2, t_1$ представляется вектором запусков $f(\sigma) = (1, 2, 2)^T$ и приводит к маркировке

$$\mu' = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ -1 & 2 & 0 \\ -1 & 1 & -1 \\ 0 & -1 & 1 \end{bmatrix} \times \begin{bmatrix} 1 \\ 2 \\ 2 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 3 \\ -1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 3 \\ 0 \\ 0 \end{bmatrix}.$$

Для определения того, является ли маркировка $\mu' = (1, 8, 0, 1)^T$ достижимой из маркировки $\mu = (1, 0, 1, 0)$, имеем линейные уравнения

$$\begin{bmatrix} 1 \\ 8 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ -1 & 2 & 0 \\ -1 & 1 & -1 \\ 0 & -1 & 1 \end{bmatrix} \cdot x, \text{ или } \begin{bmatrix} 0 & 0 & 0 \\ -1 & 2 & 0 \\ -1 & 1 & -1 \\ 0 & -1 & 1 \end{bmatrix} \cdot x = \begin{bmatrix} 0 \\ 8 \\ -1 \\ 1 \end{bmatrix}.$$

Система этих уравнений имеет решение $x^T = (0, 4, 5)$. Это соответствует разрешенной последовательности

$$\sigma = t_3, t_2, t_3, t_2, t_3, t_2, t_3, t_2, t_3.$$

Матричный подход к анализу сетей очень перспективен, но имеет и ряд недостатков. Матрица D сама по себе не полностью отражает структуру сети, так как встречные дуги между переходом и позицией (как, например, между p_1 и t_1 (см. рис. 2.15) взаимно уничтожаются в матрице D . Кроме того, в векторе запуска переходов $f(\sigma)$ отсутствует информация о последовательности запуска переходов. Это приводит к тому, что одному и тому же решению уравнения (2.1) можно поставить в соответствие несколько последовательностей запуска переходов.

В качестве положительных свойств матричного метода анализа следует отметить компактность представления информации и высокую степень формализации, облегчающую применение средств вычислительной техники.

2.6. Задание

1. Для указанного варианта построить модель ГПС сборочного типа в виде сети Петри. Получить матричное представление. Для описания функционирования использовать цветные маркеры.
2. Произвести имитационное моделирование всей ГПС или ее части (по указанию преподавателя) на ЭВМ.
3. Построить дерево достижимости.
4. Проанализировать свойства сети, доказать правильность реализации в сети алгоритма функционирования моделируемого участка.

Описание объекта моделирования

Объектом моделирования является участок ГПС сборочного типа, состоящий из сборочных центров, конвейеров для транспортировки кассет со сборочным материалом и спутников с объектами сборки, а также склада для кассет. Общая структура участка ГПС показана на рис. 2.16.

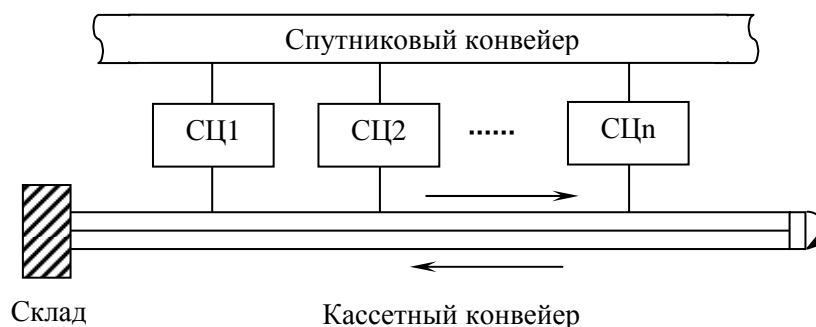


Рис. 2.16. Схема участка ГПС сборочного типа

Сборочные центры (СЦ) могут быть настроены на одну или несколько сборочных операций. По кассетному конвейеру транспортируются кассеты различных типов, адресуемые определенным сборочным центром.

Варианты заданий приведены в табл. 2.2.

ВАРИАНТЫ ЗАДАНИЙ

Таблица 2.2

№ варианта	Количество			МАРШРУТ СБОРКИ:
	СЦ	(n)	со.оп. (m)	
1	3	3	1	1оп.- СЦ1; 2оп.- СЦ2; 3оп.- СЦ3.
2	4	4	1	1оп.- СЦ1; 2оп.- СЦ2; 3оп.- СЦ3; 4оп.- СЦ4.
3	2	1	1	1оп.- СЦ1 V СЦ2.
4	3	2	1	1оп.- СЦ1; 2оп.- СЦ2 V СЦ3.
5	4	2	1	1оп.- СЦ1 V СЦ2; 2оп.- СЦ3 V СЦ4.
6	2	2	1	1оп.- СЦ1; 2оп.- СЦ2.
7	3	2	1	1оп.- СЦ1 V СЦ2; 2оп.- СЦ3.
8	4	2	1	1оп.- СЦ1 V СЦ3; 2оп.- СЦ2 V СЦ4.
9	2	3	2	1изд.(1оп.- СЦ1; 2оп.- СЦ2); 2изд.(1оп.- СЦ1).
10	3	4	2	1изд.(1оп.- СЦ1; 2оп.- СЦ2); 2изд.(1оп.- СЦ1; 2оп.- СЦ3).
11	4	3	2	1изд.(1оп.- СЦ1; 2оп.- СЦ2 V СЦ3); 2изд.(1оп.- СЦ4).
12	2	4	2	1изд.(1оп.- СЦ1; 2оп.- СЦ2); 2изд.(1оп.- СЦ1; 2оп.- СЦ2).
13	3	4	2	1изд.(1оп.- СЦ1; 2оп.- СЦ2); 2изд.(1оп.- СЦ2; 2оп.- СЦ3).
14	4	4	2	1изд.(1оп.- СЦ1; 2оп.- СЦ2); 2изд.(1оп.- СЦ3; 2оп.- СЦ4).
15	2	2	2	1изд.(1оп.- СЦ1); 2изд.(1оп.- СЦ2).
16	3	5	3	1изд.(1оп.- СЦ1; 2оп.- СЦ2; 3оп.- СЦ3); 2изд.(1оп.- СЦ1); 3изд.(1оп.- СЦ2).
17	3	4	2	1изд.(1оп.- СЦ1 V СЦ2 V СЦ3); 2изд.(1оп.- СЦ1; 2оп.- СЦ2; 3оп.- СЦ3).
18	4	4	2	1изд.(1оп.- СЦ1 V СЦ2; 2оп.- СЦ3); 2изд.(1оп.- СЦ2; 2оп.- СЦ4).
19	3	3	3	1изд.(1оп.- СЦ1 V СЦ2 V СЦ3); 2изд.(1оп.- СЦ2); 3изд.(1оп.- СЦ3).
20	3	4	3	1изд.(1оп.- СЦ1; 2оп.- СЦ2); 2изд.(1оп.- СЦ1); 3изд.(1оп.- СЦ3).

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Теория автоматического управления: Учеб. для вузов по спец. «Автоматика и телемеханика»: В 2 ч. Ч.1. Теория линейных систем автоматического управления /Н.А.Бабаков, А.А. Воронов, А.А. Воронова и др.; Под ред. А.А. Воронова. – М.: Высшая школа, 1986. – 367с.
2. Попов Е.П. Теория линейных систем автоматического регулирования и управления: Учеб. пособие для втузов. – М.: Наука; гл. ред. физ.-мат. лит., 1989. – 304 с.
3. Справочник по проектированию автоматизированного электропривода и систем управления технологическими процессами /Под ред. В.И.Круповича, Ю.Г.Барыбина, М.Л.Самовера. – М.: Энергоиздат, 1982. – 416 с.
4. Шалобанов С.В., Кочетов А.В. Методы диагностирования линейных непрерывных систем управления: Учебное пособие. – Хабаровск: Изд-во Хабар. гос. техн. ун-та, 1994. – 58 с.
5. Чернецкий В.И. Анализ точности линейных систем управления. – М.: Машиностроение, 1968. – 156 с.
6. Спиридонов Е.В., Яхонтов Ю.К. Анализ систем автоматического управления на ЭВМ. – Л.: ЛТА, 1991. – 36 с.
7. Вавилов А.А., Имаев Д.Х. Машинные методы расчета систем управления. – Л.: Изд-во Ленингр. ун-та, 1981. – 232 с.
8. Воронин В.В., Костанди Г.Г., Шалобанов С.В. Программный комплекс «Анализ топологии систем автоматического управления»// Алгоритмы и программы: Информационный бюллетень. – М.: ГосФАП, 1984. № 6 (63). С.68-69.
9. Воронин В.В., Костанди Г.Г., Шалобанов С.В. Программный комплекс «Вычисление частотных характеристик систем автоматического

управления»// Алгоритмы и программы: Информационный бюллетень.
– М.: ГосФАП, 1984. №6 (63). С.69.

10. Лазарев Ю.Ф. MATLAB 5.X. – Киев: Изд. группа BHV, 2000. – 384 с.
11. Представление и моделирование систем в структурно-матричном виде: Методические указания к лабораторной работе /Сост. В.В.Бобышев, С.В.Шалобанов. – Хабаровск, 1998. –14 с.
12. Горбатов В.А., Крылов А.В., Федоров Н.В. САПР систем логического управления. – М.: Энергоатомиздат, 1988. –232 с.
13. Питерсон Дж. Теория сетей Петри и моделирование систем. – М.: Мир, 1984. – 264 с.
14. Котов В.Е. Сети Петри. – М.: Наука, 1984. – 160 с.
15. Лазарев В.Г., Пийль Е.И. Синтез управляющих автоматов. – М.: Энергоатомиздат, 1989. – 328 с.

ОГЛАВЛЕНИЕ

Общие вопросы.....	3
1. Структурные методы цифрового моделирования систем автоматического управления	
1.1 Способы математического описания автоматических систем.....	5
1.2. Структурное цифровое моделирование САУ во временной области.....	7
1.3. Структурное цифровое моделирование САУ в частотной области.....	15
1.4. Задание.....	18
2. Применение сетей Петри для моделирования систем логического управления	
2.1. Особенности систем логического управления как объектов моделирования.....	19
2.2. Основы теории сетей Петри.....	21
2.3. Моделирование с помощью сетей Петри.....	26
2.4. Свойства сетей Петри.....	34
2.5. Методы анализа сетей Петри.....	37
2.5.1. Построение дерева достижимости.....	37
2.5.2. Метод матричных уравнений.....	41
2.6. Задание.....	45
Библиографический список.....	47

Учебное издание

Миронова Елена Ивановна

МОДЕЛИРОВАНИЕ СИСТЕМ УПРАВЛЕНИЯ

Учебно-методическое пособие

Подписано в печать _____. Тираж 20 экз.

Рязанский институт (филиал) Московского политехнического университета
390000, г. Рязань, ул. Право-Лыбедская, 26/53