

## **Курс «Вычислительные алгоритмы теории автоматического управления».**

### **Лекция 9. Методы анализа и моделирования событийно - управляемых систем.**

Событийный подход к анализу сложных систем. Простейшие математические модели событийно – управляемых систем.

Математические модели сложных систем логического управления. Инструментальные средства проектирования и анализа динамических событийно – управляемых систем.

#### **Событийный подход к анализу сложных систем.**

К сложным техническим системам относятся самолетные и космические комплексы, комплексы противовоздушной обороны, системы управления энергетическими установками, в том числе ядерными реакторами, промышленные химические, биотехнологические установки и прочее. Благодаря прогрессу микроэлектроники и программного обеспечения, в настоящее время, появились дешевые, надежные и быстродействующие микропроцессоры и ЭВМ. Это позволило строить сложные законы управления, охватывающие функционирование многих отдельных модулей системы, каждый из которых описывается своей математической моделью и своим законом управления. При этом, в основе всех этих методов, является условие (событие) попадание (или не попадание) расширенного вектора состояния сложной системы, в ту или иную область пространства состояний к определенному моменту времени. Часто событие обусловлено внешними, для системы, причинами, которые предсказать заранее трудно. Например, это может быть вертикальный порыв ветра при посадке летательного аппарата, или отказ какого – то элемента сложного комплекса.

Анализ взаимодействия законов управления различными модулями системы, особенно в случае, когда поведение других модулей, ведет к недопустимым возмущениям одного, критичного с точки зрения функционала качества, является непростой задачей. Для проведения такого анализа в последнее время появилось множество различных методов, но основным методом, позволяющим осуществить правильность выбранных технических решений, является метод комплексного имитационного моделирования.

Например, при автоматической посадке автономного БЛА на палубу корабля событие, состоящее в значительном отклонении аппарата от заданной траектории, приводит к подаче команды на прекращение посадки, которая в свою очередь приводит к изменению режима работы двигателей, изменению полетной конфигурации и заданию новой траектории полета. Однако, если не учитывать возможную траекторию БЛА после отказа от посадки, то он может столкнуться с палубными надстройками корабля, что приведет к катастрофическим последствиям. Учесть этот фактор заранее, при проектировании математической модели системы, достаточно сложно. Еще сложнее учесть динамику объекта с системой управления при различных отказах исполнительных органов и сенсоров движения.

Разработка алгоритма управления сложной системой обычно заключается в разработке соответствующего программного обеспечения для одной или нескольких бортовых ЭВМ, контролирующих различные процессы. Наличие ошибок в программном обеспечении, реализующем закон управления сложной системой, приводит к затягиванию этапов комплексной отладки и испытаний, а также к неожиданным отказам системы, как во время пробной, так и промышленной эксплуатации. Эти ошибки обусловлены различными причинами, например логической сложностью комплекса программ, взаимным влиянием программных модулей, управляющих различными контурами системы, при их совместной работе, или непредусмотренными условиями функционирования. Важно, как с точки зрения временных, ценовых, так и людских затрат, заранее, предусмотреть различные варианты поведения сложной системы, которые априори, до начала испытаний, являются не очевидными. Поэтому важную роль приобретает принцип математического моделирования таких систем.

Рассмотрим типовые математические модели, позволяющие упростить анализ таких сложных систем, а также методы их анализа.

### Простейшие математические модели событийно – управляемых систем.

#### **Релейные системы.**

Одним из простейших классов событийно – управляемых моделей являются релейные системы управления.

Рассмотрим релейную систему с линейным объектом управления:

$$\dot{x} = A \cdot x + b \cdot u, \quad x \in \mathbb{R}^n, u \in \mathbb{R}, \text{ и регулятором } u = -\text{sign}(v), \quad v = k^T \cdot x.$$

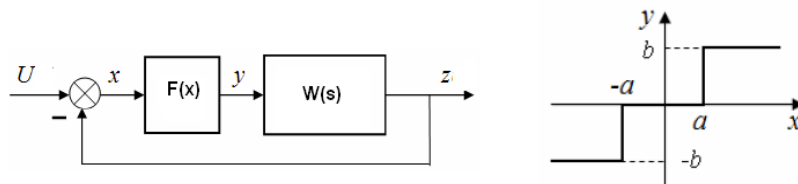
Здесь  $k \in \mathbb{R}^n$  вектор коэффициентов обратных связей. Изменение движения системы или, по сути, изменение структуры системы происходит при событии, когда вектор состояния системы удовлетворяет уравнению:  $k^T \cdot x = 0$ . То есть, имеется некоторая плоскость переключения (подпространства размерности  $(n-1)$ ) следующего вида:  $S = \{x : k^T \cdot x = 0\}$ , которая делит

фазовое пространство на области знакопостоянства сигнала  $u$ :  $\mathbb{R}^+ : k^T \cdot x < 0$   
 $\mathbb{R}^- : k^T \cdot x > 0$ . Соответственно,

$$\text{уравнения системы можно записать в следующем виде: } \dot{x} = \begin{cases} A \cdot x + b, & x \in \mathbb{R}^+ \\ A \cdot x - b, & x \in \mathbb{R}^- \end{cases}$$

#### **Пример.**

Рассмотрим релейную систему с линейным объектом управления и релейным элементом в контуре управления следующего вида.



Пусть передаточная функция объекта имеет следующий вид  $W(s) = \frac{k}{s \cdot (T \cdot s + 1)}$ .

Запишем уравнения системы:  $x = u - z$ ;  $y = F(x)$ ;  $z = W(s) \cdot y$ . Тогда, собственное движение системы (при  $u = 0$ ) можно описать с помощью следующего уравнения:  $T\ddot{x} + \dot{x} + k \cdot F(x) = 0$  или, в виде

$$\text{следующей эквивалентной системы: } \begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = -\frac{1}{T} \cdot [x_2 + k \cdot F(x_1)] \end{cases}, \text{ где } x_1 = x.$$

Построим фазовый портрет системы. Для этого, предварительно запишем уравнение изоклин:

$$\frac{dx_2}{dx_1} = -\frac{[x_2 + k \cdot F(x_1)]}{T \cdot x_2}.$$

Первый интеграл системы будет определяться следующим соотношением:

$$\frac{T}{2} \cdot x_2^2 + x_2 \cdot x_1 + k \cdot \int F(x_1) dx_1 = C,$$

где постоянная  $C$  будет определяться значениями начальных условий. В зависимости от величины  $x_1$  функция  $F(x)$  принимает следующие значения:

$$F(x) = \begin{cases} -b & x_1 \leq -a \\ 0 & |x_1| < a \\ b & x_1 \geq a \end{cases}.$$

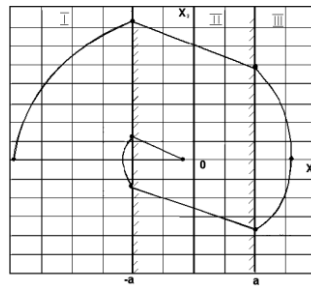
Тогда можно записать три следующих выражения, которые будут определять фазовые траектории системы на трех разных участках фазовой плоскости.

$$1 \text{ участок } (x_1 \leq -a): \frac{T}{2} \cdot x_2^2 + x_2 \cdot x_1 - k \cdot b \cdot x_1 = C;$$

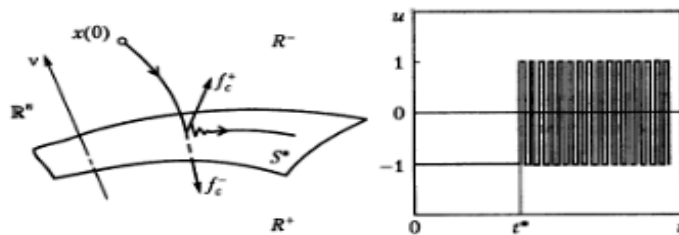
$$2 \text{ участок } (|x_1| < a): \frac{T}{2} \cdot x_2^2 + x_2 \cdot x_1 = C;$$

$$3 \text{ участок } (x_1 \geq a): \frac{T}{2} \cdot x_2^2 + x_2 \cdot x_1 + k \cdot b \cdot x_1 = C.$$

Ориентировочный фазовый портрет системы приведен на рисунке.



На некоторых участках поверхности переключения  $S$  релейные системы часто демонстрируют нетривиальное поведение – их фазовые траектории после достижения границы поверхности не переходят в другую область знакопостоянства, а следуют поверхности переключения. При этом наблюдаются быстрые переключения управляющего воздействия.



Такой режим движения релейной системы вдоль поверхности  $S$ , сопровождающийся бесконечными частыми переключениями управления называется скользящим режимом. За счет организации скользящего режима можно обеспечить системе управления нужные свойства в условиях неполной информации. Например, когда не известна точная модель объекта управления или его свойства непредвиденным образом изменяются в процессе эксплуатации.

*Условия скольжения и попадания.*

Пусть система с переменной структурой описывается уравнениями:

$$\dot{x} = F(x, u, t); x \in \mathbb{R}^n; u \in \mathbb{R},$$

где  $u = \begin{cases} u^+(x,t); s(x) > 0 \\ u^-(x,t); s(x) < 0 \end{cases}$ . Здесь  $F(x,u,t)$  непрерывная векторная функция, удовлетворяющая условию Липшица;  $u^+(x,t), u^-(x,t)$  - непрерывные функции и  $u^+(x,t) \neq u^-(x,t)$ ;  $s(x)$  - непрерывная функция, определяющая поверхность переключения. Поверхность  $S$ , определяемая уравнением  $s(x) = 0$ , делит пространство состояний  $\mathfrak{R}^n$  на два подпространства: пространство  $\mathfrak{R}_-^n = \{x; s(x) < 0\}$  и подпространство  $\mathfrak{R}_+^n = \{x; s(x) > 0\}$ .

Функция  $F(x,u,t)$ , в силу уравнения, на поверхности  $S$  терпит разрыв. Однако, при этом существуют левые и правые пределы /Ким т2 с238/:

$$F^- = \lim_{s(x) \rightarrow -0} F(x,u,t), \quad F^+ = \lim_{s(x) \rightarrow +0} F(x,u,t).$$

Производная по времени  $\dot{s}(x)$  имеет вид:

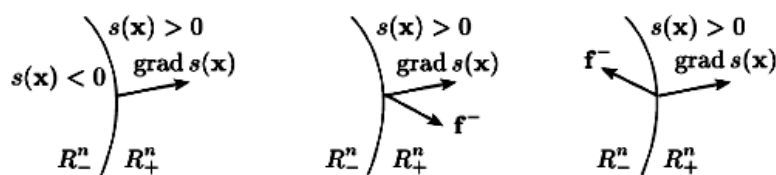
$$\dot{s}(x) = \text{grad}(s(x)) \cdot F(x,u,t).$$

Градиент  $\text{grad}(s(x))$  направлен по нормали к поверхности  $S$  в сторону возрастания функции переключения  $s(x)$ , то есть в сторону  $s(x) > 0$ . Поэтому, если предел

$$\dot{s}^- = \lim_{s(x) \rightarrow -0} \dot{s}(x) = \text{grad}(s(x)) \cdot F^-; \quad F^-(x, u^-, t), \text{ больше нуля (то есть } \dot{s}^- > 0), \text{ то угол между}$$

векторами  $\text{grad}(s(x))$  и  $F^-$  острый, то вектор  $F^-$  будет направлен в сторону подпространства  $\mathfrak{R}_+^n$ . Если же указанный предел меньше нуля ( $\dot{s}^- < 0$ ), то вектор  $F^-$  будет направлен в сторону подпространства  $\mathfrak{R}_-^n$ . Аналогично, если предел  $\dot{s}^+ = \lim_{s(x) \rightarrow +0} \dot{s}(x) = \text{grad}(s(x)) \cdot F^+; \quad F^+(x, u^+, t)$ ,

больше нуля ( $\dot{s}^+ > 0$ ), то вектор  $F^+$  будет направлен в сторону подпространства  $\mathfrak{R}_+^n$ ; если меньше нуля ( $\dot{s}^+ < 0$ ), то вектор  $F^+$  направлен в сторону  $\mathfrak{R}_-^n$ .

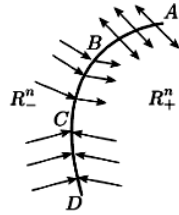


Точки на поверхности переключения  $S$  разделяются на три типа.

К первому типу относятся точки, в которых произведение  $\dot{s}^- \dot{s}^+ > 0$ , и, соответственно, векторы  $F^-$  и  $F^+$  направлены в одну сторону.

Ко второму типу относят те точки, в которых  $\dot{s}^- < 0$  и  $\dot{s}^+ > 0$ . В этих точках вектор  $F^-$  направлен в сторону подпространства  $\mathfrak{R}_-^n$ , а вектор  $F^+$  - в сторону подпространства  $\mathfrak{R}_+^n$ . То есть вектора  $F^-$  и  $F^+$  направлены в разные стороны от поверхности переключения.

Третий тип составляют точки, для которых выполняются условия  $\dot{s}^- = \text{grad}(s(x)) \cdot F^- > 0$  и  $\dot{s}^+ = \text{grad}(s(x)) \cdot F^+ < 0$ . В этом случае, векторы  $F^-$  и  $F^+$  направлены к поверхности переключения встречно друг к другу. В случае попадания изображающей точки в зону расположения этих точек на поверхности  $S$ , начинается скользящий процесс.



Условия  $\dot{s}^- = \text{grad}(s(x)) \cdot F^- > 0$ ;  $\dot{s}^+ = \text{grad}(s(x)) \cdot F^+ < 0$  называется условием существования скольжения. Указанные три типа точек, в общем случае, отделяются друг от друга многообразием, на котором, или  $\dot{s}^- = 0$ , или  $\dot{s}^+ = 0$ , или выполняются оба эти равенства.

Для того, чтобы изображающая точка, находящаяся в состоянии  $x$ ;  $s(x) \neq 0$ , двигалась в сторону поверхности скольжения, необходимо и достаточно, чтобы выполнялось соотношение  $s(x) \cdot \dot{s}(x) < 0$ . Данное условие называется условием попадания.

Проблема описания скользящего режима возникает при попадании изображающей точки при ее попадании в область скольжения на поверхности  $S$ . В этом случае функция  $F(x, u, t)$  терпит разрыв и скорость изображающей точки не равна ни  $F^-$ , ни  $F^+$ .

*Особенности решения уравнений с разрывной правой частью. Метод доопределения.*

Рассмотрим динамическую систему, которая задается отображением вида

$\Phi: \mathcal{R} \times D \rightarrow K(\mathcal{R}^n)$ . Отображение  $\Phi$  в каждой точке расширенного фазового пространства  $\mathcal{R} \times D$ ,  $t \in \mathcal{R}, x \in D \subseteq \mathcal{R}^n$  определяет многозначное поле скоростей, то есть  $\dot{x} \in \Phi(t, x)$ . Здесь  $K(\mathcal{R}^n)$  - множество всех непустых компактных выпуклых множеств из пространства  $\mathcal{R}^n$ . Такая модель является удобным аппаратом для описания неявных дифференциальных неравенств и уравнений вида:  $F(t, x(t), \dot{x}(t)) \geq 0$ , которые могут быть записаны в виде дифференциальных включений  $\dot{x} \in \Phi(t, x)$ , где мультиотображение  $\Phi$  определено как  $\Phi(t, x) = \{z : F(t, x, z) \geq 0\}$ .

Дифференциальные включения применяются и для изучения дифференциальных уравнений, правая часть которых известна лишь с некоторой степенью точности. При неоднозначности выбора управляющего воздействия уравнения системы может быть сведено также к соотношениям типа:

$$\dot{x}(t) = F(t, x(t), u(t)), \quad u(t) \in U(t, x(t)),$$

где  $U$  некоторое мультиотображение, зависящее, как от времени, так и состояния системы.

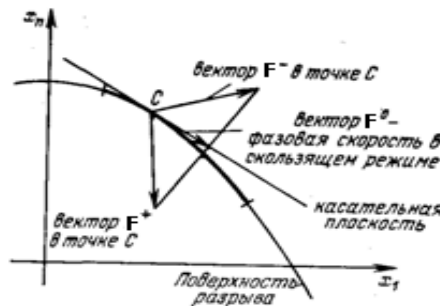
В общем случае решение задачи с начальным условием:  $\dot{x} \in \Phi(t, x)$ ,  $x(t_0) = x_0$  определено неоднозначно, поэтому приходится говорить о множестве всех решений такой задачи. Каждое решение, при этом, определено, вообще говоря, на своем промежутке времени.

Рассмотрим теперь следующую систему  $\dot{x} = F(t, x)$  с кусочно-непрерывной функцией  $F$ , заданной в области  $\mathcal{R} \times D$ ,  $t \in \mathcal{R}, x \in D \subseteq \mathcal{R}^n$  и имеющей конечное множество  $M$  меры нуль точек разрыва. Зададим мультиотображение  $\Phi(t, x)$  следующим образом. Если в точке  $(t, x)$  функция  $F$  непрерывна, то множество  $\Phi(t, x)$  будет состоять из одной точки, совпадающей со значением функции  $F$  в этой точке. Если же точка  $(t, x)$  является точкой разрыва, то множество  $\Phi(t, x)$  будет определять некоторое (по предположению выпуклое) возможное множество точек.

Тогда множество решений приведенной системы будет совпадать с решениями дифференциального включения  $\dot{x} \in \Phi(t, x)$ .

Одним из способов получения однозначного решения системы  $\dot{x} = F(t, x)$  является доопределение функции  $\Phi$  на множестве точек разрыва. Возможность существования такого доопределения для уравнений с разрывной правой частью и, наиболее распространенный, способ его осуществления был предложен Филипповым. Согласно этому доопределению, в каждой точке поверхности разрыва, вектор фазовой скорости, определяющий решение, принадлежит минимальному выпуклому замкнутому множеству, содержащему все значения функции  $F$ , когда  $x$  пробегает по  $\delta$ -окрестности рассматриваемой точки, за исключением множества нулевой меры. То есть, хотя в точках на поверхности разрыва вектор  $F$  для системы  $\dot{x} = F(t, x)$  неопределен, эти точки в некоторой окрестности составляют множество нулевой меры и могут быть исключены из рассмотрения.

Рассмотрим в качестве примера доопределения по методу Филиппова, для случая скользящего режима в скалярном случае. Чтобы определить вектор фазовой скорости в скользящем режиме, надо в каждой точке области скольжения построить векторы фазовых скоростей соответственно двух полуплоскостей существования решений системы и соединить их. Это позволяет построить минимальное выпуклое множество, о котором идет речь в доопределении по методу Филиппова. Для того, чтобы найти вектор фазовой скорости в точке разрыва, надо провести касательную к поверхности разрыва до пересечения с прямой соединяющей концы векторов  $F^+$  и  $F^-$ .



Определенное «по Филиппову» уравнение скольжения имеет вид:

$$\dot{x} = F^0(t, x), \quad F^0 = \mu \cdot F^+ + (1 - \mu) \cdot F^-, \quad 0 \leq \mu \leq 1,$$

где  $\mu$  - параметр, зависящий от взаимного расположения векторов  $F^+$ ,  $F^-$  и вектора  $\text{grad}(s(x))$  - градиента функции поверхности разрыва (переключения)  $S$ . Вычисляя параметр  $\mu$  из условия  $\text{grad}(s(x)) \cdot F^0 = 0$ , найдем следующее уравнение:

$$\dot{x} = \frac{\text{grad}(s(x)) \cdot F^-}{\text{grad}(s(x)) \cdot (F^- - F^+)} \cdot F^+ - \frac{\text{grad}(s(x)) \cdot F^+}{\text{grad}(s(x)) \cdot (F^- - F^+)} \cdot F^-.$$

Данное уравнение описывает движение системы в скользящем режиме при дополнительном предположении, что начальные условия выбираются на поверхности разрыва, то есть  $s(x(0)) = 0$ . Такое доопределение обеспечивает однозначность «сшивки» траекторий и получения решения дифференциального решения дифференциального включения  $\dot{x} \in \Phi(t, x)$  во всем области  $\mathbb{R} \times D$ ,  $t \in \mathbb{R}, x \in D \subseteq \mathbb{R}^n$ . Следует отметить, что, доопределение уравнения по методу

Филиппова и получение эквивалентного управления путем усреднения, для уравнений с разрывным (релейным) управлением (по методу Уткина) дают одинаковые уравнения движения системы по поверхностям разрыва.

*Метод эквивалентного управления.*

Вычислим производную по времени функции переключения  $s(x) = 0$  в силу уравнений движения системы  $\dot{x} = F(x, u, t)$  и приравняем ее к нулю.

То есть, будем предполагать, что точка не может покинуть: плоскость переключения  $S$ . Тогда можно записать следующее соотношение:

$$\dot{s}(x) = \frac{ds(x)}{dx} \cdot F(x, u, t) = \text{grad}(s(x)) \cdot F(x, u, t) = 0.$$

Отсюда можно найти управление  $u$ , которое называется эквивалентным управлением. Подставляя эквивалентное управление в уравнение движения системы  $\dot{x} = F(x, u, t)$ , можно построить уравнение скользящего движения.

Пусть уравнение движения объекта управления является аффинным относительно управления:

$$\dot{x} = F(x, t) + B(x, t) \cdot u,$$

где  $x \in X \subset \mathbb{R}^n$ ,  $u \in \mathbb{R}^m$ . Каждая компонента  $u_i$  вектора управления претерпевает разрывы на своей поверхности переключения, заданной уравнением:

$$s_i(x) = 0: u_i(x, t) = \begin{cases} u_i^+(x, t), & s_i(x) > 0 \\ u_i^-(x, t), & s_i(x) < 0 \end{cases}.$$

То есть имеется  $m$  поверхностей управления для каждого управления. Будем считать, что функции  $F(x, t)$  и  $B(x, t)$  вне поверхностей разрыва удовлетворяют условию Липшица. Тогда в системе такого рода могут возникать скользящие движения, которые в отличие от скалярного случая, лежат не на одной поверхности переключения, а на их пересечении. Введем в рассмотрение  $m$ -мерный вектор  $s(x) = (s_1(x), s_2(x), \dots, s_m(x))$ .

В соответствии с методом усреднения (эквивалентности) найдем уравнение для эквивалентного управления  $\bar{u}$  с учетом равенства  $\dot{s}(x) = 0$ , которое в силу уравнений системы имеет вид:

$$G \cdot F + G \cdot B \cdot \bar{u} = 0,$$

где  $G$  - матрица размерности  $(m \times n)$ , строки которой являются векторами – градиентами функций

$\dot{s}_i(x)$ , то есть  $G = (\frac{\partial s}{\partial x})$ . Предполагая, что  $\det(G \cdot B) \neq 0$  для любых  $x, t$ , отсюда найдем:

$$\bar{u} = -(G \cdot B)^{-1} \cdot G \cdot F.$$

Подставляя, полученное выражение в исходную систему получим уравнение:

$$\dot{x} = F - B \cdot (G \cdot B)^{-1} \cdot G \cdot F.$$

Данное уравнение при начальных условиях  $s(x(0)) = 0$  и принимается в качестве уравнения идеального скольжения (Уткин с46). Существенно, что в силу самого метода  $\dot{s}(x) \equiv 0$ , и, при выбранных начальных условиях, все траектории будут лежать на многообразии пересечения всех

поверхностей переключения размерности  $(n - m)$ . В соответствии с этим, вместо системы  $n$ -го порядка уравнений скольжения  $\dot{x} = F - B \cdot (G \cdot B)^{-1} \cdot G \cdot F$ , можно записать систему уравнений скольжения  $(n - m)$ -го порядка. Для этого, из системы алгебраических уравнений  $m$ -го порядка  $s(x) = 0$  необходимо выразить какие-либо  $m$  координат через остальные  $(n - m)$ . Это возможно осуществить, так как по предположению имеем  $\det(G \cdot B) \neq 0$ . Можно показать, что уравнения скользящего движения, полученные по методу эквивалентного управления, будут совпадать с уравнениями скользящего движения, полученными по методу доопределения.

*Стабилизация линейного стационарного объекта в релейной замкнутой системе.*

Рассмотрим систему с линейным объектом управления следующего вида:

$$\dot{x} = A \cdot x + b \cdot u, x \in \mathbb{R}^n, u \in \mathbb{R}, u = -\text{sign}(K^T \cdot x).$$

То есть поверхность переключения задается уравнением:

$$s(x) = K^T x; K \in \mathbb{R}^n,$$

где  $K$  - вектор обратных связей. Отсюда найдем, что  $\dot{s}(x) = K^T \dot{x} = K^T (Ax + bu)$ , и, соответственно,  $s^+ = K^T \cdot (A \cdot x + b \cdot u^+)$  и  $s^- = K^T \cdot (A \cdot x + b \cdot u^-)$ .

Тогда условия скольжения  $\dot{s}^+ = \text{grad}(s(x)) \cdot F^+ < 0$ ;  $\dot{s}^- = \text{grad}(s(x)) \cdot F^- > 0$  принимают вид:

$$s^+ = K^T \cdot (A \cdot x + b \cdot u^+) < 0, s^- = K^T \cdot (A \cdot x + b \cdot u^-) > 0.$$

Так как  $u^+ = -1$  при событии  $K^T \cdot x > 0$ , и  $u^- = 1$  при  $K^T \cdot x < 0$ , то условия возникновения скользящего режима можно записать в виде:

$$K^T \cdot b \neq \pm K^T \cdot A \cdot x, K^T \cdot x = 0.$$

Условие попадания изображающей точки фазовой траектории на поверхность скольжения имеет вид:

$$s(x) \cdot \dot{s}(x) < 0 \text{ или } K^T \cdot b > (\text{sign}(K^T \cdot x) \cdot (K^T \cdot A \cdot x) \text{ при } |K^T \cdot x| = \varepsilon,$$

где  $\varepsilon$  - некоторое малое положительное число.

Найдем эквивалентное (усредненное) управление  $\bar{u}$  при движении системы в скользящем режиме. Усредненное движение объекта под действием эквивалентного управления можно описать уравнением:

$$\dot{x} = A \cdot x + b \cdot \bar{u}.$$

Так как такое движение осуществляется по переключающей плоскости (то есть  $\dot{s}(x) = K^T \cdot \dot{x} = 0$ ), то будет справедливо равенство:

$$K^T \cdot Ax + K^T \cdot b \cdot \bar{u} = 0.$$

При этом, как было показано выше,  $K^T \cdot b \neq 0$ . Отсюда получим, что:

$$\bar{u} = -(K^T \cdot b)^{-1} \cdot K^T \cdot A \cdot x.$$

Подставив полученную формулу в уравнение усредненного движения объекта, найдем, что будет справедливо уравнение:

$$\dot{x} = [E - (b \cdot K^T) \cdot (K^T \cdot b)^{-1}] \cdot A \cdot x,$$



которое определяет движение объекта под действием эквивалентного управления по переключающей плоскости в скользящем режиме.

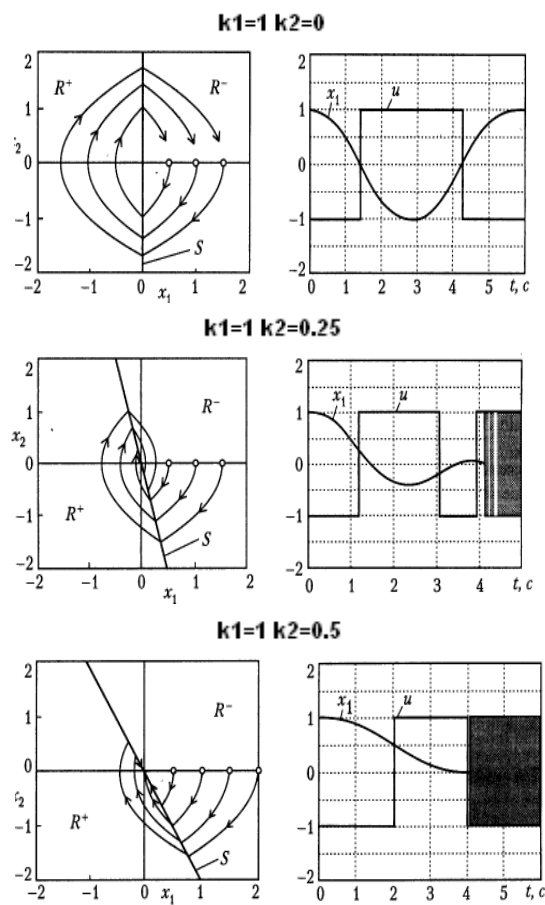
**Пример.**

Пусть задана система:

$$\begin{cases} \dot{x}_1 = x_2, \\ \dot{x}_2 = u \end{cases}$$

где  $u = -\text{sign}(v)$ ,  $v = k_1 x_1 + k_2 x_2$ . То есть линия переключения  $S$  описывается уравнением

$v = k_1 x_1 + k_2 x_2 = 0$ . Фазовые траектории системы для различных значений коэффициентов  $k_1, k_2$  приведены на рисунке.



При возникновении скользящего движения эквивалентное управление имеет  $\bar{u} = -\frac{k_1 \cdot x_2}{k_2}$ . Уравнения

скользящего движения по переключающей плоскости будут иметь вид:

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = -\frac{k_1}{k_2} \cdot x_2 \end{cases}$$

Скользящий режим будет возникать для всех плоскости переключения, для которых  $x_2 \in (-\frac{k_2}{k_1}, \frac{k_2}{k_1})$ .

### **Системы управления переменной структуры.**

Таким образом, идея принципа переменности структуры состоит в скачкообразном изменении связей между функциональными элементами регулятора в зависимости события, которое определяется фазовым состоянием замкнутой системы управления. В общем случае, такие системы называют системами управления переменной структуры (СПС). СПС можно интерпретировать как совокупность линейных подсистем и логических правил перехода от одного элемента этой совокупности к другому при пересечении фазовой точкой разделительных гиперповерхностей. Такие поверхности называют поверхностями переключения или разрыва. СПС отличается от релейных систем, прежде всего, сложностью формирования правил перехода между поверхностями переключения /Шидловский/. Если же объект или функциональные элементы являются нелинейными, то речь идет о совокупности нелинейных подсистем и, соответственно, о нелинейных многообразиях разрыва. Схема построения СПС сводится к выбору поверхностей разрыва и исходной совокупности подсистем, гарантирующих решение поставленной задачи.

Системы СПС обладают целым рядом расширенных возможностей по сравнению с системами стандартной, неизменяемой структуры. Это, прежде всего, расширенный диапазон функционирования, обусловленный более широким диапазоном возможных вариантов регулятора, между которыми система может осуществлять выбор при текущем управлении. Движение системы по области скольжения обладает свойством робастности к различным неопределенностям в модели объекта и внешним возмущающим воздействиям. Однако применение СПС ограничено имеющимися недостатками. Приведем некоторые из них /Емельянов Нов типы обр связи с124/.

1. Параметры реального скользящего режима в системе СПС зависят от «скрытых» параметров, связанных с задержками в исполнении управления и точностью оценки измеряемых параметров. Возникающие при этом «биения», то есть режимы высокочастотных колебаний, негативно сказываются на работе механических и электромеханических приводов, которые часто используются в качестве исполнительных механизмов. Это резко ограничивает практическое использование классических СПС.
2. Моделирование стандартного скользящего режима на ЭВМ возможно только методами первого порядка (Эйлера, Адамса и т.д.), так как методы более высоких порядков требуют существования производных более высоких порядков от правой части дифференциального уравнения. Это условие не выполняется для уравнений, описывающих СПС. Это делает затруднительным применение прямого цифрового управления для скользящих режимов.
3. Возникают трудности, связанные с построением дополнительных устройств наблюдения и фильтрации при наличии неопределенностей в системе, в частности связанные с получением «хороших» оценок производных выходного сигнала объекта.

### **Математические модели сложных систем логического управления.**

Рассмотренные выше релейные системы демонстрируют, как непрерывные, так и дискретные аспекты поведения. При этом, если даже математическая модель объекта является дискретной, то общий ход процесса будет нарушаться в случае срабатывания логического правила, обуславливающего попадание вектора состояния системы в некоторую область. То есть дискретность поведения замкнутой системы будет связана не с некоторым элементом дискретизации, а с логическими правилами поведения системы. При этом поведение системы, может мгновенно (с точки зрения масштаба времени исследуемого процесса) измениться, или осуществить этот процесс достаточно медленно, в масштабе времени сопоставимом с основными процессами системы. Такие системы, проявляющие свойства, как непрерывных систем, так и дискретных, со сложно вычисляемым, на основе поведения системы, интервалом дискретности, называются системами с логическим управлением.

Для описания причинно – следственных связей между событиями в таких системах часто используют логические или автоматные модели, в которых время переходных процессов можно принять равным нулю. К таким системам относят типовые системы массового обслуживания, экономические системы, многие социальные системы, системы сетевого обслуживания и прочее.

Однако, в мехатронных, впрочем, как и в химических и биологических, системах такое допущение является не допустимым. Сложная динамика механических систем требует детального изучения поведения, как отдельных элементов, так и всей системы в целом при некотором локальном воздействии. При этом реакция системы на воздействие будет не мгновенной, а растянутой, в рамках допустимой погрешности, на некоторый интервал времени. Если, во время этого переходного процесса, использовать некоторое управляющее воздействие, исходя из установившегося значения, то, в лучшем случае, можно ожидать невозможности достижения экстремально значения функционала качества, а в худшем – краха системы. В этом случае, для описания системы с логическим управлением приходится использовать модели, где в явной форме учитывается время функционирования.

### **Конечные автоматы**

Простейшим способом описать взаимодействие различных процессов и событий в системе является использование булевых функций. Если с помощью данного аппарата удастся описать логику работы системы, то ее можно представить в виде комбинационной схемы, на вход которой поступают, а с выхода снимаются бинарные сигналы, соответствующие значениям логического величины (0 или 1). Как следует из определения комбинационной схемы, значения выходных переменных  $\{Y_j\}$  в произвольный момент времени однозначно определяются значениями входных переменных  $\{X_j\}$ . Структурно комбинационная схема может быть представлена как совокупность элементарных логических схем – логических элементов, которые выполняют над входными переменными элементарные логические операции типа И-НЕ, И, ИЛИ, ИЛИ-НЕ и т.д. Число входов логического элемента соответствует числу аргументов воспроизводимой им булевой функции.

В ходе разработки комбинационных схем приходится решать задачи анализа и синтеза.

Задача анализа состоит в определении статических и динамических свойств комбинационной схемы. В статике определяются булевы функции, реализуемые комбинационной схемой по известной ей структуре. В динамике рассматривается способность надёжного функционирования схемы в переходных процессах при смене значений переменных на входах схемы, т.е. определяется наличие на выходах схемы возможных нежелательных импульсных сигналов, которые не следуют непосредственно из выражений для булевых функций, реализуемых схемой.

Задача синтеза заключается в построении из заданного набора логических элементов комбинационной схемы, реализующей заданную систему булевых функций.

Однако комбинационные схемы обладают существенным недостатком, связанным с конечным числом входных сигналов, что не позволяет достаточно достоверно описать все области изменения фазовых переменных состояния системы управления.

Поэтому, для установления причинно- следственных связей, различных событий, происходящих в сложной событийной системе, используют модели на базе конечных автоматов.

Конечный (абстрактный) автомат это устройство, характеризующееся набором внутренних состояний в которое оно попадет под воздействием команд заложенной в него программы. Переход автомата из одного состояния в другое осуществляется в определенный момент времени. Математической моделью цифрового автомата (а, в общем случае, и любого дискретного

устройства) является так называемый абстрактный автомат, определенный как набор (кортеж) /Гилл/:

$$V = (A, Q, B, \varphi, \psi),$$

где:  $A, Q, B$  - конечные множества;  $\varphi$  - функция, определенная на множестве  $Q \times A$  и принимающая значения из множества  $Q$ ;  $\psi$  - функция, определенная на множестве  $Q \times A$  и принимающая значения из множества  $B$ .

Множества  $A, Q, B$  называются соответственно входным алфавитом, алфавитом состояний и выходным алфавитом автомата  $V$ . Функция  $\varphi$  называется функцией переходов, а функция  $\psi$  функцией выходов автомата  $V$ .

Если в качестве входного и выходного алфавитов (входов и выходов автомата), а также алфавита состояния взяты декартовы произведения:

$$A = (a_1 \times a_2 \times \dots \times a_m), \quad B = (b_1 \times b_2 \times \dots \times b_p), \quad Q = (q_1 \times q_2 \times \dots \times q_n),$$

то функции:

$$\varphi: Q \times A \rightarrow Q, \quad \psi: Q \times A \rightarrow B$$

будут векторными. Соответственно автомат  $V$  будет иметь  $m$  - входов,  $p$  - выходов, а его состояние будет описываться  $n$  - мерным вектором состояния.

Рассмотрим, например, автомат  $V$ , который имеет один вход и один выход. Такой автомат работает в дискретном времени, которое принимает целые неотрицательные значения  $t = 0, 1, 2, \dots$ . В каждый момент  $t$  дискретного времени автомат находится в некотором состоянии  $q(t) \in Q$  из множества состояний автомата. Причем, предполагается, что в начальный момент  $t = 0$  он находится в начальном состоянии  $q^*$ . В момент  $t$ , будучи в состоянии  $q(t)$ , автомат воспринимает на входе букву входного алфавита  $a(t) \in A$ . В соответствии с функцией выходов  $\psi$  он выдаст в тот же момент времени  $t$  букву выходного алфавита  $b(t) = \psi(q(t), a(t))$  и, в соответствии с функцией переходов  $\varphi$ , перейдет в следующее состояние:

$$q(t+1) = \varphi(q(t), a(t)), \quad q(t+1) \in Q, \quad t = 0, 1, 2, \dots$$

На практике наибольшее распространение получили два класса автоматов - автоматы Мили (Mealy) и Мура (Moore).

Закон функционирования автомата Мили задается уравнениями:

$$q(t+1) = \varphi(q(t), a(t)), \quad b(t) = \psi(q(t), a(t)), \quad t = 0, 1, 2, \dots$$

Закон функционирования автомата Мура задается уравнениями:

$$q(t+1) = \varphi(q(t), a(t)), \quad b(t) = \psi(q(t)), \quad t = 0, 1, 2, \dots$$

Из сравнения законов функционирования видно, что, в отличие от автомата Мили, выходной сигнал в автомате Мура зависит только от текущего состояния автомата и в явном виде не зависит от входного сигнала. Для полного задания автомата Мили или Мура дополнительно к законам функционирования, необходимо указать начальное состояние и определить внутренний, входной и выходной алфавиты.

Следует отметить, что для каждого автомата Мили может быть построен эквивалентный ему автомат Мура и наоборот.

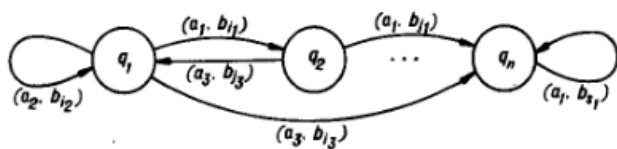
Одним из основных способов задания конечных автоматов является непосредственное перечисление элементов множеств  $A, Q, B$  и задание функций  $\varphi, \psi$  с помощью прямоугольных таблиц. Каждая строка таблицы, определяющей функции  $\varphi$  или  $\psi$ , взаимно однозначно сопоставлена некоторому символу  $a_i$  из алфавита  $A$ , а каждый столбец сопоставлен однозначно символу  $q_j$  алфавита  $Q$ .

	$q$				
$a$	$q_1$	$q_2 \dots$	$q_j \dots$	$q_n$	
$a_1$					
$a_2$					
$\vdots$					
$a_i$			$\varphi(q_j, a_i)$		
$\vdots$					
$a_m$					

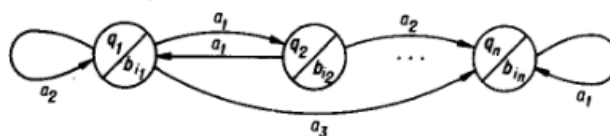
	$q$				
$a$	$q_1$	$q_2 \dots$	$q_j \dots$	$q_n$	
$a_1$					
$a_2$					
$\vdots$					
$a_i$			$\psi(q_j, a_i)$		
$\vdots$					
$a_m$					

Другим способом задания конечных автоматов являются диаграммы Мура, которые используются при небольшой размерности  $(m \times p \times n)$ .

Для построения диаграмм Мура на плоскости размещаем  $n$  кругов и, внутри каждого круга, запишем символ  $q_j$  алфавита  $Q$ . Далее рассмотрим всевозможные пары  $(a_i, q_j)$ ,  $a_i \in A, q_j \in Q$ . Для каждой такой пары от круга, в котором записан символ  $q_j$ , проведем ориентированную дугу к кругу, в которой записан символ  $\varphi(q_j, a_i)$ . Этой дуге назначим пару значений  $(a_i, \varphi(q_j, a_i))$ . Полученное в результате изображение и называется диаграммой Мура автомата  $V$ . Легко заметить, что из каждого круга выходит ровно  $m$  дуг. Для автомата Мура дуги обозначаются только первыми элементами пары, так как вторые элементы для всех исходящих дуг будут одинаковыми. Поэтому значение второго элемента из множества  $B$  записывают внутри круга, из которого выходят ориентированные дуги.



Автомат Мили



Автомат Мура

Пусть  $V = (A, Q, B, \varphi, \psi)$  - конечный абстрактный автомат. Для каждого состояния  $q$  автомата  $V$  можно рассмотреть набор  $(A, Q, B, \varphi, \psi, q^0)$ , определяющий автомат  $V_q$  с выделенным начальным состоянием  $q^0$ . Автомат  $V_q = (A, Q, B, \varphi, \psi, q^0)$  называется инициальным абстрактным автоматом. На диаграммах Мура начальное состояние инициального автомата обозначается звездочкой, располагаемой рядом с соответствующим этому состоянию кругом.

Рассмотренные выше абстрактные автоматы можно разделить на следующие типы:

- 1) полностью определенные и частичные;
- 2) детерминированные и вероятностные;
- 3) синхронные и асинхронные;

Полностью определенным называется абстрактный конечный автомат, у которого функция переходов и функция выходов определены для всех пар  $(q_j, a_i)$ ,  $q_j \in Q$ ,  $a_i \in A$ .

Частичным называется абстрактный автомат, у которого функция переходов или функция выходов, или обе эти функции определены не для всех пар  $(q_j, a_i)$ ,  $q_j \in Q$ ,  $a_i \in A$ .

К детерминированным относятся автоматы, у которых выполнено условие однозначности переходов: автомат, находящийся в некотором состоянии  $q_j \in Q$ , под действием любого входного сигнала  $a_i \in A$  не может перейти более, чем в одно состояние. Если это условие нарушается, то это будет вероятностный автомат, в котором при заданном состоянии  $q_j \in Q$  и заданном входном сигнале  $a_i \in A$  возможен переход с заданной вероятностью в различные состояния.

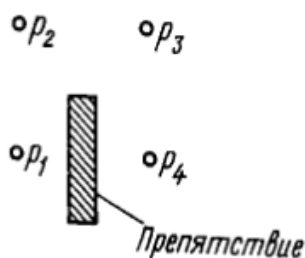
Для определения синхронных и асинхронных автоматов вводится понятие устойчивого состояния.

Состояние  $q_s \in Q$  автомата называется устойчивым, если для любого состояния  $q_j \in Q$  и любого входного сигнала  $a_i \in A$  таких, что выполняется равенство  $\varphi(q_j, a_i) = q_s$ , имеет место  $\varphi(q_s, a_i) = q_s$ . То есть состояние устойчиво, если попав в это состояние под действием некоторого сигнала  $a_i \in A$ , автомат выйдет из него только под действием другого входного сигнала  $a_k \in A$ ,  $a_k \neq a_i$ .

Автомат, у которого все состояния устойчивы, называется асинхронным. Автомат называется синхронным, если он не является асинхронным.

#### Пример.

Рассмотрим робот - манипулятор с позиционно – контурной системой управления /Ющенко с357/. При этом область достижимости схвата манипулятора содержит, в общем случае, бесконечно много точек позиционирования. Манипулятор может перемещаться по некоторой программе, которая задается в виде последовательности точек позиционирования. Приход схвата в заданную точку фиксируется сенсорной системой (например, видеокамерой) и подтверждается соответствующим программным сигналом. Перевод манипулятора из некоторой точки в соседние точки перемещения осуществляется по прямолинейной траектории. Рассмотрим автомат Мили, который осуществляет реализацию некоторой программы работы манипулятора в ограниченном пространстве.



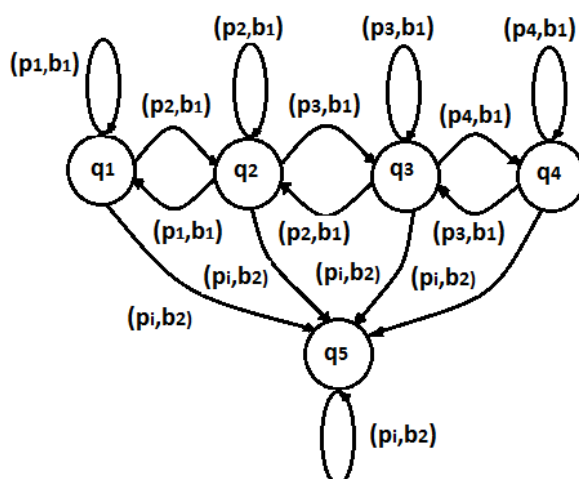
Обозначим последовательность точек программы как:  $P = \{p_i\}, i = 1, 2, \dots, 4$ . Тогда множество состояний манипулятора можно описать как  $Q = \{q_1, q_2, q_3, q_4, q_5\}$ , где  $q_i = p_i, i = 1, 2, \dots, 4$ ,  $q_5$  - положение манипулятора не совпадает с последовательностью точек заданной траектории. Соответственно

входной алфавит будет определяться, как множество  $A = \{p_1, p_2, p_3, p_4\}$ . Алфавит выходов автомата будет характеризоваться множеством из двух точек  $B = \{b_1, b_2\}$ , где  $b_1$  - успешный перевод манипулятора в заданную точку траектории,  $b_2$  - сбой в переводе манипулятора в заданную точку траекторию.

Переходная функция  $\varphi$  переходов состояния и выходная функция  $\psi$  будут задаваться следующими таблицами:

Функция $\varphi$						Функция $\psi$					
$A$	$Q$					$A$	$Q$				
	$q_1$	$q_2$	$q_3$	$q_4$	$q_5$		$q_1$	$q_2$	$q_3$	$q_4$	$q_5$
$p_1$	$q_1$	$q_1$	$q_5$	$q_5$	$q_5$	$p_1$	$b_1$	$b_1$	$b_2$	$b_2$	$b_2$
$p_2$	$q_2$	$q_2$	$q_2$	$q_5$	$q_5$	$p_2$	$b_1$	$b_1$	$b_1$	$b_2$	$b_2$
$p_3$	$q_5$	$q_3$	$q_3$	$q_3$	$q_5$	$p_3$	$b_2$	$b_1$	$b_1$	$b_1$	$b_2$
$p_4$	$q_5$	$q_5$	$q_4$	$q_4$	$q_5$	$p_4$	$b_2$	$b_2$	$b_1$	$b_1$	$b_2$

Диаграмма Мура такого автомата будет иметь следующий вид.



Имеется большое количество различных программных средств, позволяющих реализовывать конечные автоматы в виде соответствующего программного кода. Например, switch – технология представляет достаточно удобные средства для реализации сетей конечных автоматов /Шальто/.

### Сети Петри

Другим мощным аналитическим инструментом исследования систем логического управления являются сеть Петри, которая представляет собой модель описания потоков событий в системе.

Сеть Петри состоит из четырех основных элементов: множества позиций  $P = \{p_1, p_2, \dots, p_n\}$ , множества переходов  $T = \{t_1, t_2, \dots, t_m\}$ , входной функции  $I: T \rightarrow P^\infty$ , которая отображает каждый переход  $t_j \in T$  во множество  $I(t_j)$  входных позиций перехода, и выходной функции  $O$ , которая отображает переход в множество позиций  $O(t_j)$ , называемых выходными позициями перехода /Питерсон с15/.

Понятие «событие» обычно интерпретируется, как изменение какой – либо компоненты ситуации или процесса. Связь между переходами или позициями определяется с помощью условий, каждое из которых имеет всего два значения: истина (выполнено) или ложь (не выполнено).



Переход может наступить, если выполнены все условия, от которого зависит его наступление. Если переход наступил, то изменяются значения некоторых условий. Связи между позициями (условиями) и переходами (событиями) в модели Петри изображаются в виде двудольного ориентированного мультиграфа с двумя типами вершин: кружками и полочками, и кратными дугами. Кружки или позиции соответствуют условиям, а полочки переходы – событиям. Позиции и переходы соединяются дугами.

Если позиция (условие)  $p_j$  является входной для перехода (события)  $t_i$ , то дуга направлена от  $p_j$  к  $t_i$ . Если позиция  $p_j$  является выходной позицией перехода  $t_i$ , то дуга направлена от  $t_i$  к  $p_j$ .

Если условие выполнено, то внутри кружка помещается маркер (точка), и, в этом случае говорят, что условие маркировано или размечено /Варшавский с34/. Для сетей Петри общего вида допускается наличие более одной метки в позиции.

Достоинством языка сетей Петри является то, что он позволяет описывать параллельные процессы, а также имеет средства для описания и разрешения конфликтных состояний, когда необходимо запретить одновременное развитие нескольких процессов /Котов/.

Формально сеть Петри определяется следующим образом.

Сеть Петри  $V$  называется набор из пяти элементов вида  $V = \langle P, T, I, O, M_0 \rangle$ , где:

$P = \{p_1, p_2, \dots, p_n\}$  - конечное непустое множество позиций или условий;

$T = \{t_1, t_2, \dots, t_m\}$  - конечное непустое множество переходов или событий;

$I: T \rightarrow P^\infty$  - является входной функцией: отображает переходы  $t_j \in T, j = 1, 2, \dots, m$  в комплекты входных позиций (условий)  $I(t_j)$ ;

$O: T \rightarrow P^\infty$  - является выходной функцией: отображает переходы  $t_j \in T, j = 1, 2, \dots, m$  в комплекты выходных позиций (условий)  $O(t_j)$ ;

$M_0: P \rightarrow \{0, 1, 2, \dots\}$  - начальная разметка сети  $V$ .

Следует учитывать, что множество позиций (условий) и множество переходов (позиций) не пересекается, то есть  $P \cap T = \emptyset$ .

Маркировка сети Петри  $V$  задается в виде векторной функции  $M: P \rightarrow \{0, 1, 2, \dots\}$ , которая определяет число меток в каждой позиции сети. То есть маркировка в каждый такт времени задается вектором  $M = (\mu_1, \mu_2, \dots, \mu_n)^T, n = |P|$ . Распределение меток в позициях сети Петри определяет порядок ее выполнения, который зависит от последовательности реализации переходов. Переход (событие) может быть реализован (разрешен), только если он становится активным, то есть когда каждая из его входных позиций содержит число меток не меньшее, чем число дуг, соединяющих ее с этим переходом. Когда переход имеет несколько выходных позиций, после его реализации все они получают метки, то есть произойдет распараллеливание процесса.

В обычных (оригинальных) сетях Петри предполагается, что реализация перехода рассматривается как мгновенное событие, занимающее нулевое время.

*Пример.*

Рассмотрим непомеченную сеть Петри  $V = \langle P, T, I, O \rangle$ , заданную следующей структурой:

$P = \{p_1, p_2, p_3, p_4, p_5\}$ ;



$$T = \{t_1, t_2, t_3, t_4\};$$

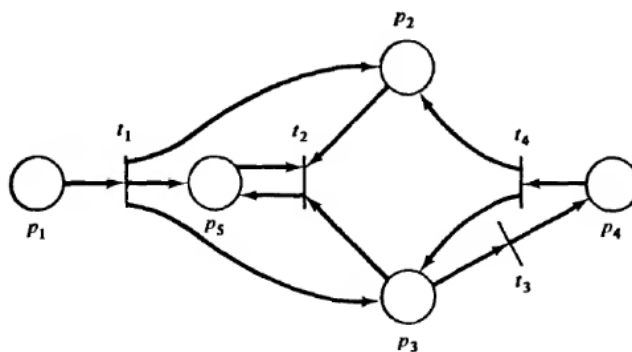
$$I(t_1) = \{p_1\}, O(t_1) = \{p_1, p_3, p_5\};$$

$$I(t_2) = \{p_2, p_3, p_5\}, O(t_2) = \{p_5\};$$

$$I(t_3) = \{p_3\}, O(t_3) = \{p_4\};$$

$$I(t_4) = \{p_4\}, O(t_4) = \{p_2, p_3\}.$$

Граф сети Петри, эквивалентный заданной структуре имеет следующий вид.



Если структуру сети  $V$  задать как:

$$P = \{p_1, p_2, p_3, p_4, p_5, p_6\};$$

$$T = \{t_1, t_2, t_3, t_4, t_5\};$$

$$I(t_1) = \{p_1\}, O(t_1) = \{p_2, p_3\};$$

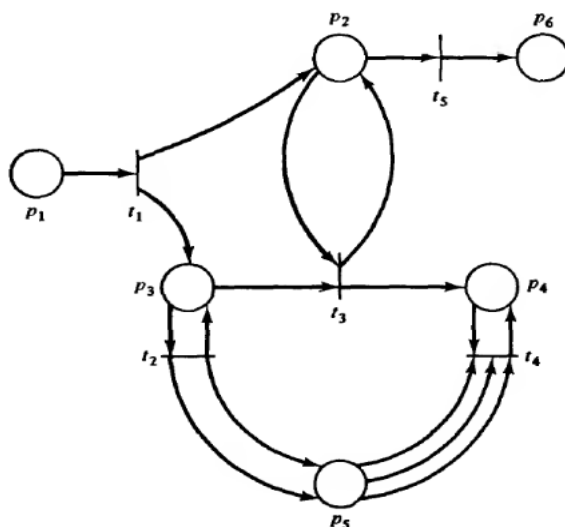
$$I(t_2) = \{p_3\}, O(t_2) = \{p_3, p_5, p_5\};$$

$$I(t_3) = \{p_2, p_3\}, O(t_3) = \{p_2, p_4\};$$

$$I(t_4) = \{p_4, p_5, p_5, p_5\}, O(t_4) = \{p_4\}.$$

$$I(t_5) = \{p_2\}, O(t_5) = \{p_6\},$$

то графическое представление сети имеет следующее изображение:

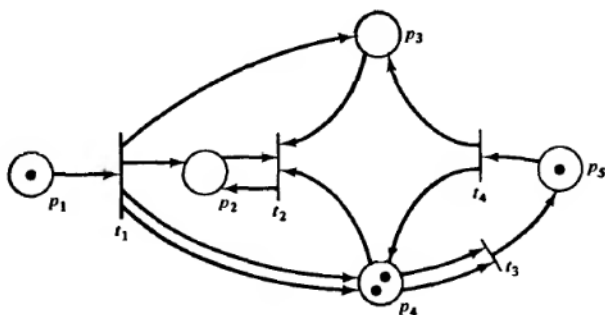


Выполнением сети Петри управляют количество и распределение маркеров в сети. Сеть Петри выполняется посредством запусков переходов. Каждый разрешенный переход запускается удалением маркеров из его входных позиций и образованием новых маркеров, помещаемых в его

выходные позиции. Как уже отмечалось, переход запускается удалением разрешающих маркеров из его входных позиций и последующим помещением в каждую из его выходных позиций по одному маркеру для каждой дуги. Запуски переходов могут осуществляться до тех пор, пока существует хотя бы один разрешенный переход. Когда не останется ни одного разрешенного перехода, выполнение переходов и работа сети Петри прекращается.

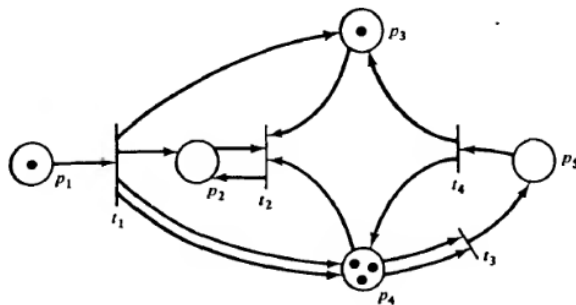
#### Пример.

Рассмотрим маркированную сеть Петри, в которой разрешены переходы  $t_1, t_3, t_4$ . Вектор начальной маркировки равен  $M_0 = (1, 0, 0, 2, 1)^T$ .



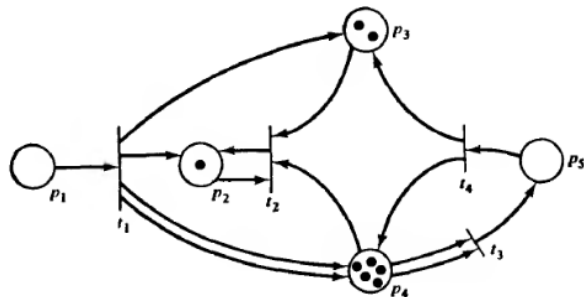
Маркировка, полученная в результате запуска перехода  $t_4$ , будет определяться вектором

$M_1 = (1, 0, 1, 3, 0)^T$ , и имеет изображение:



Маркировка, полученная в результате запуска перехода  $t_1$ , будет определяться вектором

$M_1 = (0, 1, 2, 5, 0)^T$ , и имеет изображение:



Таким образом, при выполнении сети Петри возникают две связанные последовательности – реализуемых переходов и маркировок  $M_0, M_1, \dots$ . Сеть Петри называется ограниченной, если все ее позиции имеют ограниченное число маркеров.

Следует отметить, что сеть Петри представляет собой модель структуры управления, но не отражает связь управляющего блока с объектом управления. Поэтому появились различные расширения сети Петри, которые позволяют моделировать взаимодействие сети с управляемым процессом. Например, в одном из подходов в сеть Петри вводилось специальное множество позиций, каждая из которых представляет собой отдельный входной или выходной символ, характеризующий условие взаимодействия с объектом. В позицию, соответствующую входному символу, метка помещается извне, а не в результате реализации переходов сети. Метка, появившаяся при выполнении сети Петри в позиции, соответствующей выходному символу, удаляется после выполнения действий с объектом управления /Питерсон/.

### *Разновидности сетей Петри*

Оригинальные сети Петри предназначены для моделирования логики параллельных асинхронных процессов без учета реального времени срабатывания переходов. Моделирование таким переходом событие называют примитивным. Однако, такие сети, даже с примитивными переходами, представляют очень удобное средство моделирования систем с логическим управлением. Поэтому непримитивные события часто представляют в виде двух примитивных событий, соответствующих началу и концу реального события, и условий, которые принимают истинное значение, когда такое событие происходит. Такие сети называются сетями Петри с предикатами на переходах. Активизированный переход может быть реализован только при истинном значении сопоставленного с ним предиката. При этом выполняется соответствующее переходу событие (или действие) и создается новая маркировка сети Петри.

Во временных сетях Петри каждому переходу  $t_j \in T, j = 1, 2, \dots, m$  ставится пара величин  $0 \leq a_j \leq b_j$ . Если переход  $t_j$  активизирован в момент времени  $q$ , то он не может быть реализован ранее, чем в момент  $q + a_j$ , и должен быть реализован до или в момент  $q + b_j$ . Такие временные сети используются для удаленного управления объектами с помощью вычислительных сетей. Если временной интервал срабатывания перехода  $t_j$  является случайным, то такие сети называются стохастическими временными сетями Петри.

В ряде случаев, необходимо при работе сети Петри оценить различные типы условий или ресурсов, которые характеризуют состояние системы. Для этого используют раскрашенные сети Петри с цветными маркерами. Так что одна позиция (условие) сети Петри может содержать маркеры разных цветов. Тогда для активизации перехода его входные дуги должны иметь метки с определенными цветами, соответствующим цветам маркеров.

### **Инструментальные средства проектирования и анализа динамических событийно – управляемых систем.**

Описание динамического объекта управления обычно осуществляется с помощью моделей другой, не событийной природы, например, с помощью дифференциальных уравнений, структура которых и коэффициенты зависят от внешних событий. При этом требования к системе управления таким объектом задаются в виде соответствующих ограничений на фазовые координаты и их временные закономерности. Это приводит к серьезным затруднениям при проектировании сложных систем, так как язык описания объекта не совпадает с языком описания требований или спецификаций к системе. Одним из инструментов решения этой проблемы является супервизорное управление, которое предполагает создание центрального блока, отвечающего за согласование задач управления объектом с координацией автономных компонент системы /Cassandras/.

При дискретно – событийном подходе к построению блока логического контроля событиями в комплексной системе управления выделяют два основных этапа. На первом этапе составляется перечень основных процессов, происходящих в системе. На основании словесного описания каждого процесса, схемы связей между входами и выходами, а также графа переходов строится семейство взаимодействующих автоматов Мура – Мили (или сетей Петри). Затем на основе обобщенной схемы взаимодействия полученных автоматов строится основной автомат (супервизор), управляющий автоматной сетью. Таким образом, логическая (автоматная) система управления представляет собой иерархическую схему взаимодействующих детерминированных конечных автоматов вида:

$$V = \{V_0, V_{1,1}, \dots, V_{1,k_1}, V_{2,1}, \dots, V_{r,1}, \dots, V_{r,k_r}\},$$

где  $r$  - количество уровней иерархии автоматов;  $k_i, i = 1, 2, \dots, r$  - число автоматов на  $i$  - ом уровне иерархии.

Автомат  $V_0$  называется основным, а остальные автоматы вложенными. Автомат  $V_{i,j}$  может вызывать (передавать управление) стоящему ниже по иерархии автомату  $V_{i+1,j}, j = 1, 2, \dots, k_i$ .

Автоматная схема  $V$  получает от объекта управления события, характеризующие изменения его состояния, а также сама оказывает воздействия на объект. Кроме того, происходит взаимодействие автоматов между собой, которое находится под контролем главного автомата (супервизора)  $V_0$ .

Поэтому множество событий объекта управления необходимо дополнить событиями, связанными с изменениями состояния схемы  $V$ .

Множество кортежей всех возможных событий системы управления имеет вид:

$$E = (\{e_{0,1}, e_{0,1}, \dots, e_{0,k_0}\}, \dots, \{e_{i,1}, \dots, e_{i,k_i}\}, \dots, \{e_{r,1}, \dots, e_{r,k_r}\}), k_0 + k_1 + \dots + k_r = s.$$

Входные сигналы, поступающие на автоматную схему  $V$  с объекта управления, опишем с помощью множества кортежей:

$$A = (\{a_{0,1}, a_{0,1}, \dots, a_{0,l_0}\}, \dots, \{a_{i,1}, \dots, a_{i,l_i}\}, \dots, \{a_{r,1}, \dots, a_{r,l_r}\}), l_0 + l_1 + \dots + l_r = m.$$

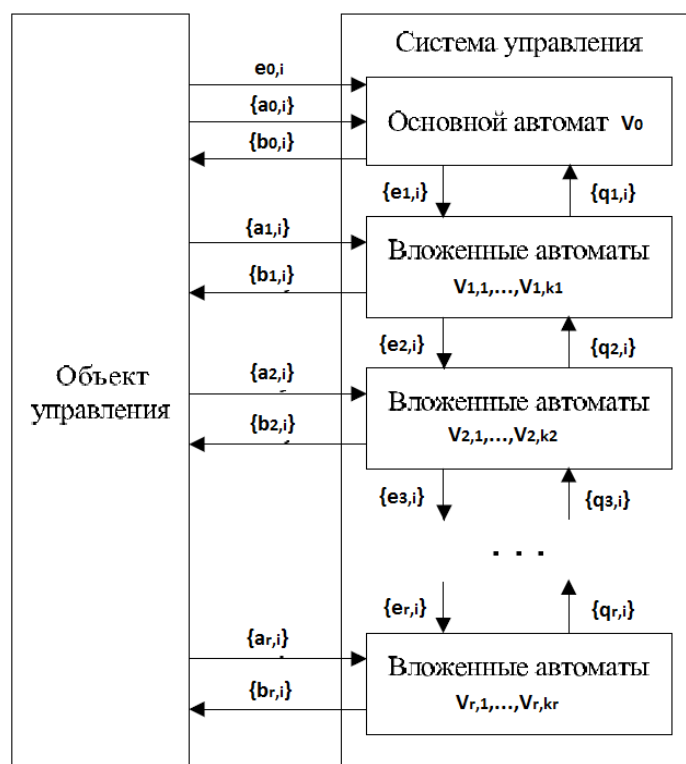
Выходные сигналы (управление), поступающие с автоматной схемы  $V$  на объект управления будем задавать с помощью множества кортежей:

$$B = (\{b_{0,1}, b_{0,1}, \dots, b_{0,j_0}\}, \dots, \{b_{i,1}, \dots, b_{i,j_i}\}, \dots, \{b_{r,1}, \dots, b_{r,j_r}\}), j_0 + j_1 + \dots + j_r = p.$$

Состояние автоматной схемы  $V$  будем характеризовать множеством кортежей:

$$Q = (\{q_{0,1}, q_{0,1}, \dots, q_{0,h_0}\}, \dots, \{q_{i,1}, \dots, q_{i,h_i}\}, \dots, \{q_{r,1}, \dots, q_{r,h_r}\}), h_0 + h_1 + \dots + h_r = n$$

Тогда модель взаимодействия логической системы управления с объектом управления можно изобразить следующей диаграммой.



При осуществлении события, супервизор распознает его на основе анализа сигналов на выходах объекта управления, а также учитывает состояния вложенных автоматов, управляющих отдельными компонентами объекта управления. Затем супервизор производит последовательность выходных воздействий, которые необходимо совершить при попадании вектора координат системы в новое состояние, и, только после этого, переходит в новое состояние /Кузьмин/. При этом выходное воздействие, представляющее собой передачу управления с распознанным событием вложенному автомату, считается выполненным только лишь после реакции вложенного автомата на это событие. До тех пор, пока такое выходное воздействие не осуществится, работа супервизора приостанавливается.

Так как ординарный конечный автомат, в общем случае, не предназначен для управления объектом при разных событиях, в том числе при временных ограничениях. Поэтому для синтеза супервизора чаще всего используют временные раскрашенные предикатные сети Петри /Hruz/.

#### Пример супервизорной дискретно – событийной системы управления.

Рассмотрим систему моделирования ядерного реактора, которая включает в себя модель ядерного реактора с автоматическим управлением и контролем параметров реакции /Сытник/.

Спецификации отражают следующие требования к функциям системы:

- оперативно реагировать на изменения условий протекания реакции путем выработки необходимых корректирующих воздействий;
- гарантировать поддержание всех важных параметров в допустимых пределах;
- при выходе параметров за пределы нормального диапазона попытаться восстановить нормальный режим работы;
- если параметры работы реактора таковы, что ситуация не может быть исправлена, произвести экстренный останов реактора.

Состав системы управления:

- блок контроля параметров ядерного реактора;
- блок запуска реактора в работу;

- блок остановки работы реактора;
- блок защиты.

Основные функции взаимодействия блоков системы:

- блок контроля через интерфейс реактора получает данные о состоянии реактора (или от его модели);
- если произошло аварийное событие (реактор находится в одном из критических состояний), то управление передается блоку защиты;
- блок контроля на основе данных о параметрах реактора формирует выходное количественное воздействие – «поднять» или «опустить» стержни.

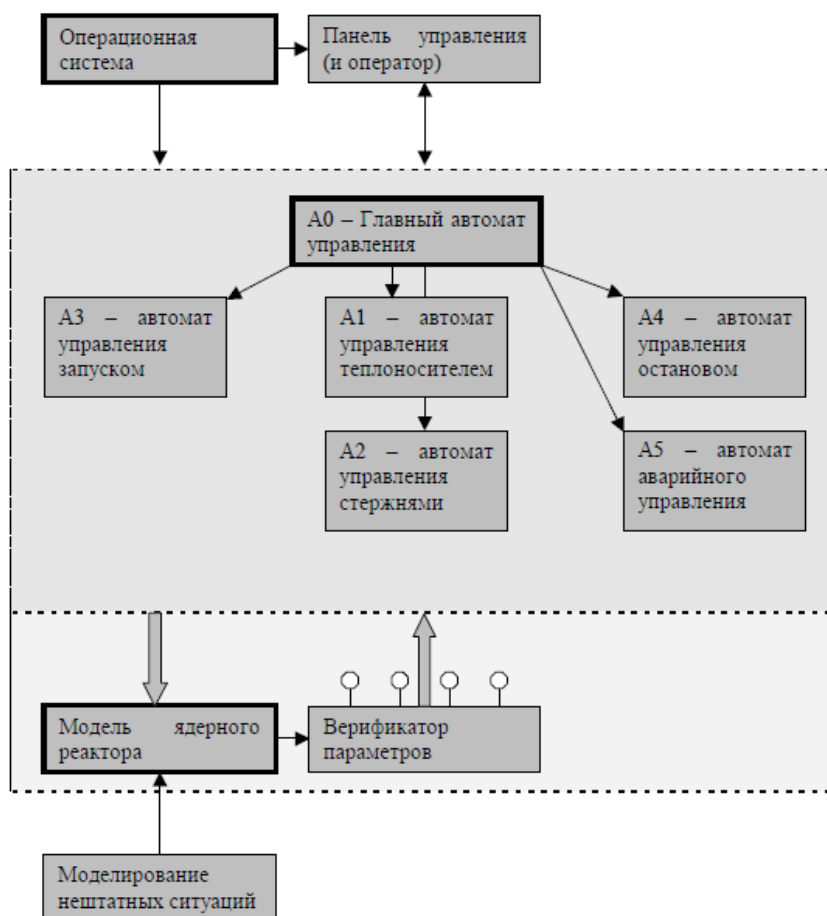
Система управления имеет возможность изменять глубину погружения стержней в активную зону, а также изменять скорость циркуляции теплоносителя. В свою очередь параметры реактора зависят от положения стержней и скорости теплоносителя. Таким образом, входными параметрами модели реактора являются:

- текущее время работы реактора;
- глубина погружения стержней в активную зону;
- скорость подачи теплоносителя.

Выходные параметры модели реактора описываются значениями следующих величин:

- коэффициент размножения нейтронов;
- поток нейтронов активной зоны;
- температура теплоносителя;
- температура рабочей зоны;
- полная тепловая мощность реактора;
- полезная мощность (электрическая) реактора.

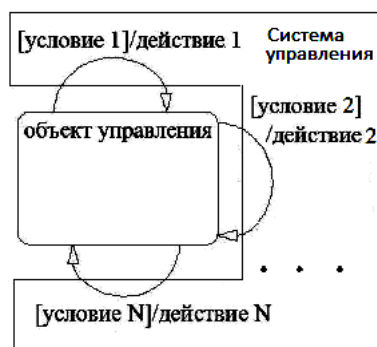
Блок – схема системы имеет вид, показанный на рисунке.



### Модели пакета StateFlow Toolbox Matlab

Одним из широко распространенных программных средств для моделирования дискретно – событийных систем является пакет Stateflow среды Matlab. Пакет реализует визуальную технологию программирования гибридных автоматов, которая является расширением теории конечных автоматов и, несмотря на появление новых инструментальных технологий, продолжает оставаться весьма популярным, среди разработчиков. Это связано его тесными связями, как с пакетом Simulink, так и пакетом Real Time Workshop. который позволяет создавать файлы на языках C/C++ по построенной схеме моделирования системы /Дьяконов Идентификация с388/.

Простейший гибридный автомат представляется в виде направленного графа, имеющего графическое представление, показанное ниже.



Единственная вершина графа гибридного автомата отображает динамику объекта управления в фазе ожидания системы (между моментами ее срабатывания). Грани графа задают переходы. С вершиной сопоставляется модель объекта управления (набор начальных состояний и дифференциальных уравнений для непрерывного динамического объекта, набор начальных состояний и разностных уравнений для дискретного динамического объекта и прочее). Грани – это фаза реагирования. Каждой грани ставится в соответствие условие перехода к этой фазе и действие, осуществляемое системой в этой фазе /Рогачев/.

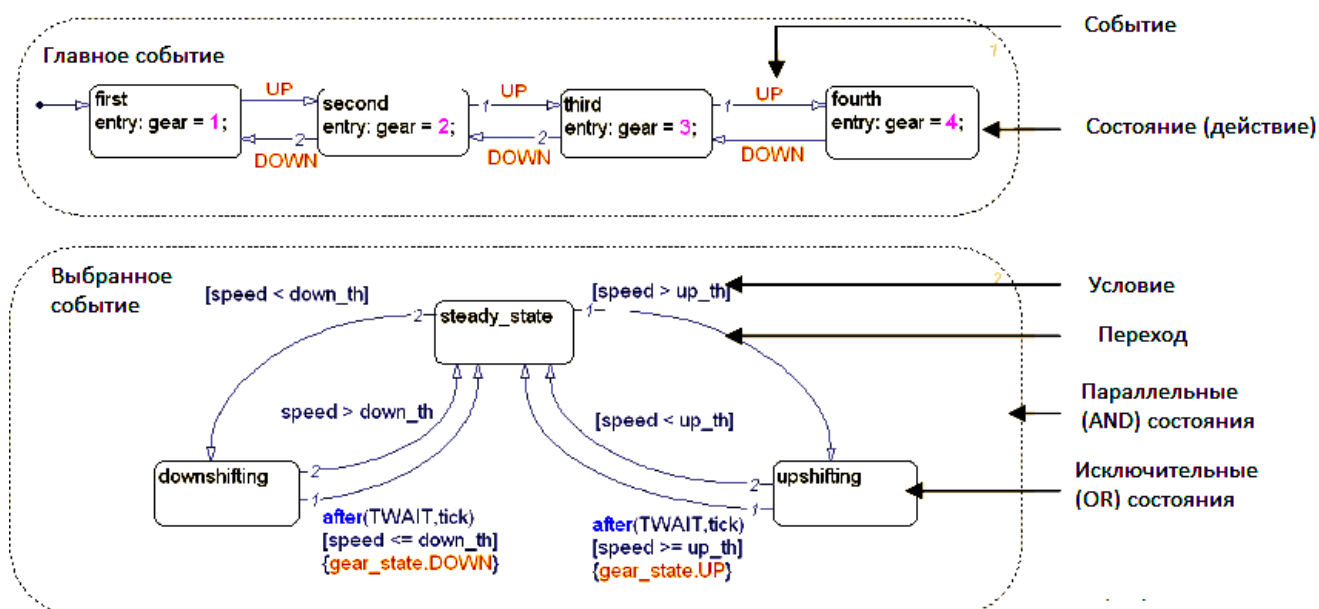
Основой пакета Stateflow является SF – диаграммы, создаваемые графическими средствами пакета, на основе анализа спецификаций (требований) к системе. Главные неграфические компоненты таких диаграмм - это события, действия и данные (числовые переменные), а основные графические компоненты - состояние и переход. Событие рассматривается как некоторая сущность, появляющаяся из – за явлений, происходящих вне рассматриваемой системы, и которое, возможно, требует некоторых ответных действий. События, при этом, считаются мгновенными. В системе Stateflow обычно понимают термин «состояние», как условия, в которых моделируемая система пребывает некоторое время, и в течение которого она ведет себя одинаковым образом.

Под действиями понимается реакция моделируемой системы на события. Подобно событиям, действия принято считать мгновенными.

В графическом представлении диаграммы переходов состояния представлены прямоугольными полями. Под термином «переход» понимают изменение состояния, обычно вызываемое некоторым значительным событием. Как правило, состояние соответствует промежутку времени между двумя такими событиями. Переходы показываются в диаграммах переходов линиями со стрелками, указывающими направление перехода. Каждому переходу могут быть сопоставлены условия, при выполнении которых переход осуществляется. С каждым переходом и каждым состоянием могут быть соотнесены некоторые действия. Действия могут дополнительно обозначаться как действия, выполняемые однократно при входе в состояние; действия,

выполняемые многократно внутри некоторого состояния и действия, выполняемые однократно при выходе из состояния.

На рисунке изображена типичная SF – диаграмма некой абстрактной системы /[Stateflow User Guide 2015/](#).



Основными объектами SF – диаграмм являются следующие типы:

- графический объект (состояние, действие) в виде прямоугольника с округленными углами ; каждое состояние имеет состояние – родителя и может иметь дочерние состояния, которое изображается вложенным в родительское; состояния могут нейтральными, занятыми, параллельными, то есть существующими одновременно (AND) и взаимно исключающими друг друга (OR); если состояние является единственным, то его родителем является сама SF – диаграмма; состояние может иметь признак памяти, то есть функция имеющая приоритет в выполнении;
- дуги (переходы), которые отражают изменение состояние системы и представляются обычно красными стрелками ; переходы имеют метки (маркеры) которые описывают условия, при которых происходит переход из одного места диаграммы в другое; для указания альтернативных путей перехода системы из одного состояния в другое служат признаки альтернативы, которые упрощают построение SF – диаграмм; альтернативный путь представляется в виде окружностей, имеющих стрелку перехода;
- событие не является графическим объектом, так как определяется с помощью того или иного условия; однако события имеют свойства, одним из которых является область видимости.

Другим объектом, который не отражается на графических диаграммах Stateflow, является процедура, которая реализуется с помощью моделей конечных автоматов: модели Мура, связующей процедуру с состояниями, и модели Мили, связующей процедуру с переходами. При записи процедур используется специальная семантика пакета Stateflow.

Понятие «данные» представляет в SF – моделях числовые значения и непосредственно на диаграмме не указываются.

*Пример.*





*объектно-ориентированная концепция представления моделей предметной области в форме классов, обладающих структурными свойствами и поведением.*

**Определение.** *Класс - это абстракция совокупности реальных объектов, которые имеют общий набор свойств и обладают одинаковым поведением.*

Каждый объект в этом случае рассматривается как экземпляр соответствующего класса. Объекты, которые не имеют полностью одинаковых свойств или не обладают одинаковым поведением, по определению не могут быть отнесены к одному классу. Важной особенностью классов является возможность их организации в виде некоторой иерархической структуры, которая по внешнему виду напоминает схему классификации понятий формальной логики. Иерархия понятий строится следующим образом. В качестве наиболее общего понятия или категории выбирается понятие, имеющее наибольший объем и, соответственно, наименьшее содержание. Это самый высший уровень абстракции для рассматриваемой иерархии.

В рассмотренных языках объектно – ориентированного моделирования (ООМ) можно выделить несколько характерных особенностей, существенных для моделирования сложных динамических систем. При моделировании непрерывных систем в качестве интерфейса объекта рассматривается совокупность внешних переменных, а не совокупность методов и сообщений, как в языках объектно-ориентированного программирования. Это отражает объективные особенности функционирования непрерывных систем. Следовательно, в языке моделирования сложных динамических систем, понятие активного объекта должно быть расширено.

Под объектом понимается структурный компонент моделируемой системы, функционирующий параллельно в модельном времени с другими структурными компонентами. Существующие языки ООМ ориентированы в значительной степени на технологию "промышленного" моделирования, предполагающую преимущественное использование в конкретных проектах заранее разработанных прикладных библиотек классов. Предполагается, что определения классов в основном будут писаться высококвалифицированными разработчиками библиотек, а обычные пользователи будут строить модель в виде совокупности стандартных объектов, взаимодействующих явно через связи или неявно через уравнения или дискретные действия объекта-контейнера. Наиболее близким, в этом смысле, является подход, используемый в пакете Simulink Matlab. Однако, далеко не для всех приложений, такой подход является естественным представлением модели в виде структурной схемы.

В исследовательских задачах, в учебном процессе и на ранних стадиях разработки систем управления пользователь в основном имеет дело с моделями отдельных изолированных систем. Не случайно, для этих задач чаще используются математические пакеты, ориентированные на изолированные системы и функциональный стиль описания (например пакет Control System Toolbox, Robust Control System Toolbox и прочее), а не пакеты компонентного моделирования. Для моделирования изолированных систем актуальным является формирование сложного поведения из отдельных фрагментов. Такие модели далее становятся основой для разработки компонентов, которые будут объединены в модель всей системы.

Возможность пользователя разрабатывать укрупненные высокоуровневые компоненты, отражающие естественную структуру прикладной задачи, является одним из важнейших направлений развития современных инструментов моделирования. Поэтому формализм гибридного автомата позволяет строить сложные поведения путем последовательного, а не параллельного включения отдельных компонентов. Очевидно, что динамические системы, приписываемые состояниям гибридного автомата, можно рассматривать как компоненты или экземпляры соответствующих классов. Таким образом, открывается своего рода "новое измерение" для ООМ. Используя возможность последовательного включения компонентов, можно многие задачи

моделирования решать вообще без составления структурных схем, ограничиваясь моделью изолированной системы.

Практически во всех существующих языках ООМ предполагается жесткая фиксация стереотипа атрибутов, который затем невозможно изменить в производном классе. Однако опыт объектно-ориентированного программирования говорит о том, что использования наследования и полиморфизма объектов действительно чрезвычайно плодотворны, если вся иерархия классов тщательно продумана заранее. Но в ходе исследовательского моделирования сделать это практически невозможно. Прежде всего, это относится к определению стереотипа переменных, т. е. разделению переменных на входы, выходы, контакты, параметры и прочее. На ранних этапах проектирования или научного исследования, на стадии отработки отдельных компонентов чрезвычайно трудно предугадать возможные взаимодействия, которые могут потребоваться при объединении этих компонентов на последующих этапах.

### **Язык UML.**

Язык UML 2.0 ориентирован для применения в качестве языка моделирования различными пользователями для решения широкого класса задач объектно – ориентированного проектирования. При этом термин "унифицированный" в названии UML 2.0 не является случайным и имеет два аспекта. С одной стороны, он фактически устраняет многие из несущественных различий между известными ранее языками моделирования и методиками построения диаграмм. С другой стороны, создает предпосылки для унификации различных моделей и этапов их разработки для широкого класса систем, не только программного обеспечения, но и проектных процессов. Семантика языка UML 2.0 определена таким образом, что она не является препятствием для последующих усовершенствований при появлении новых концепций моделирования. В настоящее время на рынке CASE-средств представлено множество программных инструментов, поддерживающих нотацию языка UML 2.0 и обеспечивающих интеграцию, включая прямую и обратную генерацию кода программ, с наиболее распространенными языками и средами программирования, такими как C++, C#, Java, Visual Basic и прочее.

Язык UML 2.0, также как и его предыдущие версии, предназначен для решения следующих задач:

- предоставления в распоряжение всех пользователей легко воспринимаемую и выразительную нотацию для визуального моделирования, специально предназначенную для разработки и документирования моделей сложных систем самого различного целевого назначения.
- обеспечения интуитивной ясности и понятности основных конструкций языка моделирования;
- обеспечения интеграции различных представлений системы (изолированных непрерывных моделей, конечных автоматов, сетей Петри, гибридных автоматов и прочее) в новую парадигму в методологии объектно – ориентированного моделирования, практическим следствием и центральным стержнем которой является язык UML 2.0;
- снабжения исходных понятий языка UML 2.0 возможностью расширения и специализации для более точного представления моделей систем в конкретной предметной области.
- обеспечения языка UML 2.0 механизмом расширения базовых понятий, который является самостоятельным элементом и имеет собственное описание в форме правил расширения; при этом все пользователи должны быть способны строить модели систем для большинства обычных приложений с использованием только базовых конструкций языка UML 2.0 без применения механизма расширения; новые понятия и графические обозначения целесообразно применять только в тех ситуациях, когда имеющихся базовых конструкций явно недостаточно для построения моделей системы.