

НОВАЯ ПАРАДИГМА ЛОГИЧЕСКОГО УПРАВЛЕНИЯ

Бокр Й.

Западнoчешский университет в г. Пльзене, Чехия

э. адрес: bokr @ kiv. zcu. cz

В статье рассматривается динамический логический объект, объясняются причины его „динамики“, вводится его собственная траектория состояний. Критически анализируется имеющаяся концепция логического управления. Показывается, что система логического управления динамическим объектом представляет собой каноническую композицию. Вводится потенциально-динамический объект, строится на котором система логического управления. Статья также занимается программным, оптимальным и адаптивным логическим управлением.

Ключевые слова: динамический объект, собственная траектория состояний, система автоматического логического управления, каноническая композиция, потенциально-динамический объект, программное управление, оптимальное и адаптивное управление.

Пролог

Автор будет признателен всем читателям, которые ему любезно сообщат не только о его формальных, а, прежде всего, о принципиальных сомнениях.

1. Введение

Пусть требуется выполнение некоторой траектории состояний в заданном динамическом логическом объекте.

Если динамический объект после своего запуска выполняет „самопроизвольно“ желаемую траекторию состояний, переводя самого себя из своего начального состояния в некоторое из своих достижимых заданных конечных (целевых) состояний, то говорят о директивном управлении объектом. Упомянутую траекторию состояний в динамическом объекте естественно назвать его **собственной траекторией**.

Если нужно динамический объект, находящийся в распоряжении заказчика, заставить выполнять не собственную траекторию состояний, а траекторию принужденную, которая представляет собой некоторую траекторию, состоящую из отрезков собственных траекторий в объекте, существует ли в объекте такая, то заданный динамический объект входит в состав системы автоматического логического управления с обратной связью (канонической композиции) и ищется подходящий, управляющий автомат.

Если объекты управления не суть динамические, а потенциально-динамические, представляют собой которые подавляющее большинство объектов управления, то подбирают управляющий автомат так, чтобы добиться выполнения требуемой собственной траектории состояний в создаваемой динамической системе управления. Разумеется, что

возникшую динамическую систему управления можно включить в состав иерархически выше стоящей системы управления и заставить ее выполнять принужденную траекторию состояний, опять-таки, существует ли она.

Функционирование динамических логических объектов рассматривается, на состоятельном уровне детализации.

Поскольку, проектируя управляющие автоматы действительных объектов управления, разработчики встречаются с управляющими автоматами, алфавиты которых содержат сотни букв, то разрабатывались языки (отметим некоторые из них; например, начиная с языка графических схем алгоритмов, продолжая через АЛОС, анкетный язык ТАЯ и ЯРУС, закончим языком временных диаграмм Горбатова) и разрабатываются (FBD – диаграмма функциональных блоков, IL – язык инструкций, ST – высший язык программирования, LD – лестничные схемы и Графсет) языки для простого описания функционирования управляющих автоматов, включая их простой синтез и реализуемых прежде всего программируемыми логическими контроллерами.

Откровенно говоря, имеющееся описание функционирования управляющего автомата, напоминает которое скорее „копию“ описания работы объекта управления, невообразимо трудоемко; поэтому и написана эта статья.

2 Динамический логический объект

Выделяя [1] некоторые моменты t_0, t_1, t_2, \dots реального времени, в котором рассматриваются состояния действительного динамического объекта, предполагается, что последовательности смен состояний и изменений воздействий на объект в моменты t_0, t_1, \dots (траектории состояний) достаточно полно описывают действие объекта. Смена состояний объекта или изменение воздействия на объект относятся к временному интервалу $[t_i, t_{i+1})$ или к окрестности момента t_i ($i = 0, 1, \dots$). При описании поведения объекта существенными являются не сами значения моментов t_i , а лишь их ход (течение). Во многих случаях оказывается возможным считать количество состояний конечным (быть может, весьма значительным). Выбор множества состояний объекта в общем случае решается интуитивно в зависимости от степени знания объекта, часто прибегая к случайным пробам и исправляемым ошибкам, ибо не существуют общие правила выбора множества состояний.

Принимаем [2] в качестве эмпирически установленного факта, что одни смены состояний от момента к моменту мы способны описать как будто они детерминированы (очередное состояние достоверно определяется заданием исходного состояния и воздействия в предыдущий момент), в то время, как другие смены мы можем описать так, как будто являются недетерминированными (значения актуальных состояний и воздействия еще не позволяют однозначно определить состояние к следующему моменту, а лишь судить о возможных очередных состояниях). Под физическим источником недетерминизма подразумевают воздействие на объект **неявных**, т.е. неизмеримых, неконтролируемых возмущений.

О п р е д е л е н и е 2. 1. Под **измеримыми (явными) возмущениями** будем понимать датчиками измеренные, неуправляемые параметры, характеризующие протекание технологического процесса в объекте (как например температура, давление, скорость, подачи, число оборотов, некоторые решения субъекта и т.п.), причем их значение в любой момент времени, за исключением момента воздействия возмущения на объект, не известно, т.е. оценки возмущений встречаются случайно.

Если оценка возмущений не своевременна или физически не осуществима или слишком дорого обходится, то такие возмущения считают неявными.

Одиночные явные возмущения до их воздействия на объект суть случайны, т.е. ожидается воздействие любого из них; воздействует ли актуально явное возмущение на объект, оно в данный момент времени становится достоверным

Пример 2.1. : Пусть заданы бинарные алфавиты возмущений $Z_i = \{z_i, \bar{z}_i\}$, где $i = 1, 2, 3$, так, что $z_1 = t_i \leq T_i$, $\bar{z}_1 = t_i > T_i$; $z_2 = (сч := m)$, $\bar{z}_2 = (сч < m)$; z_3, \bar{z}_3 , причем t_i – температура и T_i – требуемая температура детали, $сч$ - содержание счетчика деталей и z_3 - деталь находится в соответствующей позиции. Отсюда алфавит возмущений $Z = \times_{i=1}^3 Z_i = \{z_1^{\sigma_1} z_2^{\sigma_2} z_3^{\sigma_3}\}_{\sigma_1, \sigma_2, \sigma_3 \in \{0,1\}^3}$ при $z_i^{\sigma_i} = z_i \sigma_i \vee \bar{z}_i \bar{\sigma}_i$. Построить соответствующий фрагмент собственной траектории состояний (рис.2.1. б)) при заданных иерархических разбиениях на Z и на $Z - \{\bar{z}_3\}$ (рис.2.1.а)). Заметим, что экономную запись, например $(z_1 \vee z_2) \bar{z}_3$, нельзя толковать так, что на объект

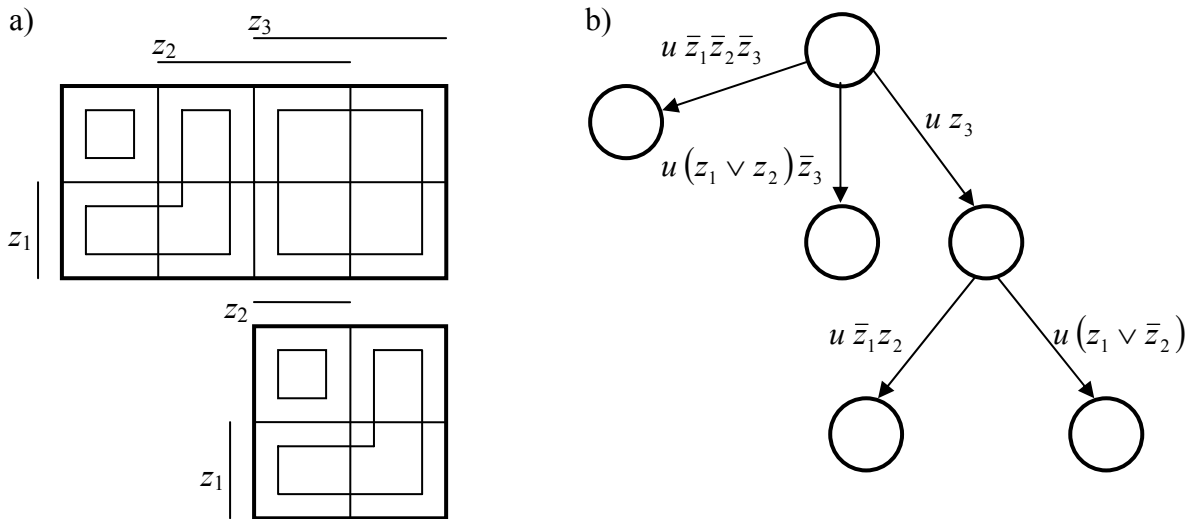


Рис. 2.1. а) Разбиение на Z и на $Z - \{\bar{z}_3\}$, б) фрагмент разбиения соответствующей собственной траектории состояний из примера 2.1.

воздействуют возможные возмущения z_3 совместно с z_1 или с z_2 , а возможные возмущения $z_1 z_2 \bar{z}_3$ или $z_1 \bar{z}_2 \bar{z}_3$ или $\bar{z}_1 z_2 \bar{z}_3$. ■

Все возмущения, воздействующие на объект, должны быть независимыми, т.е. воздействие любого возмущения не является причиной воздействия другого возмущения.

Наблюдение 2.1. Наблюдаем ли мы за очередным состоянием перехода между состояниями, то объект непременно находился в предыдущий момент в исходном состоянии перехода. Находится ли объект в исходном состоянии и воздействует ли на объект стимул, то объект всегда перейдет в очередное состояние.)*

)* Пусть имеется парусник, в направлении продольной оси которого может дуть ветер в смысле с кормы до носа, причем один корабль с убранными парусами не оказывает аэродинамическое сопротивление. Плывет ли парусник, то достоверно дует ветер, т.е. ветер является необходимой и, тем самым, исполнительной причиной судоходства. Дует ли ветер и паруса поставлены, то парусник всегда плывет, т.е. поставление парусов является достаточной и, тем самым, иницирующей причиной судоходства.

Т е о р е м а 2.1. Исходное состояние перехода между состояниями динамического объекта является необходимой причиной и входное воздействие есть достаточная причина состоянческого перехода.

Отметим, что теорема 2.1. – непосредственное следствие наблюдения 2.1. – противоречит утверждению о единственной, т.е. необходимой и достаточной, причине перехода между состояниями в объекте, является которой по [3] входное воздействие на объект.

Н а б л ю д е н и е 2.2. Всякий переход между состояниями надо завести и исполнить. Завести переход не представляет проблему; в системе управления имеются цепочки: датчик состояний объекта управления → усилитель → исполнительный механизм (актор), исполняет который физически переход между состояниями в объекте управления.

Т е о р е м а 2.2. Необходимая и достаточная причина перехода между состояниями есть соответственно причиной исполнительной и инициирующей.

О п р е д е л е н и е 2.2. Логический объект (рис.2.2.) является **динамическим объектом**, моделируется ли конечным автоматом)* Мура

$$O = \langle U \times [Z], S, Y, \delta_O, \Lambda_O, s_H \rangle, \tag{1}$$

где U, Z, S, Y – алфавит соответственно входной (управления), явных (измеримых, контролируемых) возмущений, состояний, выходов, δ_O – отношение

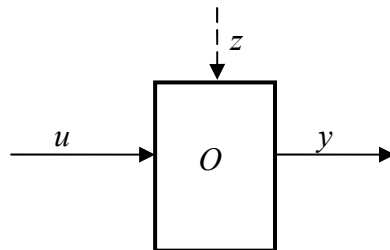


Рис. 2.2. Блок-схема логического объекта.

$$\delta_O : S \times U \times S : \langle s, u, s' \rangle,$$

в частности функция

$$\delta_O : S \times U \times [Z] \rightarrow S : \langle s, u, [z] \rangle \mapsto s' \text{)**}$$

)* Модель есть отражение своего объекта, а объект является манифестацией своей модели с тем, что названия квантовых сигналов входных, возмущающих, состоянческих и выходных образуют алфавит соответственно входной, возмущений, состояний и выходной.

)** Все ожидаемые случайные воздействия на объект возмущений z_i образуют – $Z = \bigvee_{i=1}^{|Z|} z_i$ - достоверное возмущение при $z_i z_j = \emptyset$ ($i \neq j$), где \emptyset – невозможное возмущение. Отсюда, если $\delta_O(s, u, z_1) = \delta_O(s, u, z_2) = \dots = \delta_O(s, u, z_{|Z|}) = s'$ то $\delta_O(s, u, Z) = \delta_O(s, u) = s'$.

переходов при $[Z] = Z / \{e\}$ и $[z] = z / e$ (e – пустое слово) с тем, что $\delta_O(s, u, e) = \delta_O(s, u) = s'$, Λ_O – функция выходов

$$\Lambda_O : S \rightarrow Y : s \mapsto y$$

и s_H – начальное состояние. Если функция Λ_O выходов проста, то O называют **наблюдаемым**; т.е. $s = \Lambda_O^{-1}(y)$.

Определение 2.3. Состояние s_H или s_{Ki} ($i=1,2,\dots,k$) называется соответственно **управляемым** в s_{Ki} или **достижимым** из s_H , существует ли управляющее слово $u_{j1} u_{j2} \dots u_{jf}$ из U^f , возможно совместно с возмущающим словом $z_{j1} z_{j2} \dots z_{jf}$ из Z^f , такое, что

$$T_i = (\{s_H\} \times U \times S) \circ (S \times U \times S) \circ \dots \circ (S \times U \times \{s_{Ki}\}_i) = \{s_H\} \times U^f \times \{s_{Ki}\}_i, \quad (2)$$

в частности

$$T_i = (\{s_H\} \times U \times [Z] \rightarrow S) \circ (S \times U \times [Z] \rightarrow S) \circ \dots \circ (S \times U \times [Z] \rightarrow \{s_{Ki}\}_i) = \{s_H\} \times (U \times [Z])^f \rightarrow \{s_{Ki}\}_i \quad (3)$$

где \circ – оператор композиции отношений, спец. функций, $\delta_O(s_H, ______, s_H)$, спец. $\delta_O(s_H, ______) = s_H$, где $______$ – пробел ($______ \in U$) и s_{Ki} – некоторое конечное состояние, т.е. $\delta_O(s_{i,f-1}, u_{if}, s_{Ki})$ и $\delta_O(s_{Ki}, u_{if}, s_{Ki})$, спец. $\delta_O(s_{i,f-1}, u_{if}, [z_{if}]) = \delta_O(s_{Ki}, u_{if}, [z_{if}]) = s_{Ki}$ и разумно ожидать также $\delta_O(s_{Ki}, ______, s_{Ki})$, спец. $s_{Ki} = \delta_O(s_{Ki}, ______, [z_{if}])$.

Определение 2.4. **Собственной** или **принужденной** траекторией состояний в объекте O (1) называется последовательность сопряженных состоянческих переходов T_i (2), спец (3), такая, что соответственно $u_{j1} = u_{j2} = \dots = u_{jf} = u$ или хотя бы $u_{jk} \neq u_{jl}$ ($k \neq l$; $k, l = 1, 2, \dots, f$).

Встречаются ли в траектории состояний T_i состоянческие циклы, то, из-за технологических соображений, ни начальное s_H , ни конечное s_{Ki} состояния не входят ни в какой цикл состояний, а все состоянческие циклы в любой траектории состояний выполняются только в конечное число раз; репозиция траекторий, как правило, обеспечивается.)*

Считают, что необходимо удовлетворить метаконечно-автоматное требование: **любая собственная траектория состояний не может быть повторно начата до ее завершения!**

Следуя теоремам 2.1. и 2.2. очевидно, что собственные траектории состояний не выполняются в динамическом объекте стихийно под воздействием лишь управления (возможно вместе с возмущениями), а вполне законно, ибо необходимой и, тем самым, исполнительной причиной любого перехода сопряженных переходов собственной траектории является исходное состояние каждого из переходов траектории.

Предполагается, что состояния динамического объекта изменяются скачкообразно (дискретно) в моменты t_i, t_{i+1} , относятся ли переходы между состояниями либо к временным

)* Хотелось бы назвать собственные траектории состояний фундаментальными или Хаффмена; но, поскольку могут траектории содержать состоянческие циклы, то упомянутыми названиями нельзя, строго говоря, воспользоваться.

интервалам соответственно $[t_i, t_{i+1})$, $[t_{i+1}, t_{i+2})$, либо в окрестностях моментов соответственно t_i , t_{i+1} . Это означает, что в каждом из состояний объект проводит (находится, пребывает) конечное время соответственно $\Delta t_i = t_{i+1} - t_i$ или $\Delta t_{i+1} = t_{i+2} - t_{i+1}$ и затем скачком (мгновенно) переходит в состояние-преемника. Поскольку реальный переход из одного наблюдаемого состояния в другое занимает определенное время и реальное промежуточное состояние при этом не наблюдается, отождествляется промежуточное состояние с состоянием конечного автомата либо реалистически с исходным, либо фиктивно с его преемником (см. таблицу переходов), и тогда действительную продолжительность перехода можно „включить“ во время пребывания в исходном состоянии, а сам конечно-автоматный переход считать мгновенным.

Пример 2.2. Построить при помощи ориентированных сплошных линий собственные траектории состояний T_i , ($i = 1, 2, 3, 4$) в таблице переходов (табл.2.1.), автомата $\langle \{a, b, c, d\} \times \{z_1, z_2\}, \{j\}_{j=1}^8, \delta_0, 1 \rangle$. ■

Табл. 2.1.

s	uz	s'				
		a	b	c	dz_1	dz_2
1		2 ↓	↙ 2 ↘	2 ↓	↓ 2	2 ↓
2		↓ 3	↓ 3, 6	↓ 3 ↓	↓ 3	↓ 3 ↓
3		↓ 4	↓ 4 ↓	↓ 4 ↓	↓ 4 ↓	↓ 4 ↓
4		↓ 5	↓ 5 ↓	↓ 5 ↓	↓ 5 ↓	↓ 5 ↓
5		↓ 6	↓ (5) ↓	↓ 2, 6 ↓	↓ 6 ↓	↓ 6 ↓
6		↓ 7	↓ 7 ↓	↓ 7 ↓	↓ 8	↓ 7 ↓
7		↓ 8	↓ 8 ↓	↓ 8 ↓	- ↓	↓ 4 ↓
8		(8) ↓	(8) ↓	(8) ↓	(8) ↓	-

О п р е д е л е н и е 2.4. Динамический объект назовем **управляемым**, существует ли в объекте траектория состояний, ведущая из начального состояния в требуемые состояния конечные.

3. Логическое управление по Глушкову

Говоря о логическом управлении, постулируется, что:

- системы управления, реализуют которые [4] алгоритмы управления, есть либо без задержек во времени (комбинационные схемы), либо с заданной разработчику заказчиком временной задержкой (временные логические схемы первого и второго типов), либо схемой, строится которая в виде временной схемы третьего типа, когда разработчик вводит задержки в обратные связи,
- при выборе источников информации (*ИИ*) обычно, сознательно или интуитивно руководствуются [5] принципом максимального воздействия *ИИ* на автоматизированный процесс; иными словами, в первую очередь выбирают такие *ИИ*, изменение состояния которых наиболее явно и непосредственно сказывается на изменении состояний исполнительных механизмов,
- в общем, естественно считать управляющий автомат последовательностным автоматом Мили [6],

- само собой разумеется, что управляющий автоиат детерминированный автомат Мура [7],
- управляющий автомат это ледетерминированный автомат Мура или Мили [8,9],
- управляющий автомат - детерминированный или недетерминированный полуавтомат [10],
- в [11,12,13] говорится о последовательно управляемых машинах или о последовательностной выработке их управления,
- управляющая часть представляет, само собой разумеется, последовательностную логическую цепь [14],
- управляющая часть должна по здравому смыслу обладать внутренней памятью [15], хотя эффективные методы проектирования управляющих автоматов игнорируют необходимость введения дополнительных внутренних переменных, ибо они присутствуют в схеме в неявном виде как внутренняя память ячеек, представляющих операторы языка сигнальных графов,
- предположение $|Q|=1$ (Q – алфавит состояний управляющего автомата) является, по всей вероятности, слышком сильной предпосылкой [16],
- управляющий автомат это не временной объект [17], а замкнутый объект, изменяющий свои состояния (осуществляющий переходы между состояниями) под влиянием тех или иных событий, объединяемых в общем случае в булевы формулы, равенство единице каждой из которых инициирует соответствующий переход между состояниями, включая сохранение состояния.

Известно, что понимание, ставшее уже классическим, логического управления по Глушкову [18], предложившему модель системы автоматического логического управления, состоящую из кооперирующих динамического управляющего и операционного автоматов, последний из которых соответствует объекту управления, считают, вне всякого сомнения, корректной.

Пусть динамический объект управления естественно наблюдаем и управляем и пусть его конечно-автоматной моделью является автомат Мура во избежание, т.н. цикла мгновенной зависимости (см. приложение 1.). В [1] применяется детерминированный автомат, в [20] имеют в виду детерминированные процессы, протекающие в системе управления, в [6,7,14, 21] используется недетерминированный автомат, функция переходов которого имеет вид

$$\delta_o : S \times U \rightarrow 2^S : \langle s, u \rangle \mapsto \mathcal{S}',$$

где $\mathcal{S}' \subseteq S$ и по умолчанию $\exists! s'$ ($s' \in \mathcal{S}'$). В [9] строится граф схема функционирования объекта управления, причем выходные двоичные переменные и операторы записываются в вершины соответственно условные и операторные. В [4,5,8,11.12,13,17] объекты управления суть наборы вентиляторов, заслонок, захлопок, клапанов, электродвигателей и т.п. и в зависимости от их свойств и связей обладают объектами различным поведением, которое должно быть учтено перед разработкой системы управления, построив заранее технологическую схему объекта управления. В [22] говорится: если для формализации операционных объектов управления необходимо развитие бесконечных автоматов, то для управляющих автоматов возможно применение теории конечных автоматов.

Значит, при моделировании объектов управления, в отличие от моделей управляющих автоматов, не встречаются, как правило, конечные автоматы. Поэтому, против ожидания пусть упорядоченная шестерка

$$O = \langle U, S, Y, \delta_o, \Lambda_o, s_H \rangle. \quad (4)$$

где U, S, Y – алфавит соответственно управления, состояний, выходной, δ_O – отношение

$$\delta_O : S \times U \times S : \langle s, u, s' \rangle,$$

в частности функция

$$\delta_O : S \times U \rightarrow S : \langle s, u \rangle \mapsto s',$$

переходов, Λ_O – функция

$$\Lambda_O : S \rightarrow Y : s \mapsto y$$

выходов и s_H – начальное состояние, является моделью объекта управления.

Управляющий автомат, входящий в состав системы автоматического логического управления – см. рис. 3.1.a), моделируется конечным автоматом с практически не встречающимся возмущающим входом либо Мили, либо Мура

$$A = \langle Y \times [Z], Q, U, \delta_A, \lambda_A / \Lambda_A, q_H \rangle, \quad (5)$$

где Z, Q – алфавит соответственно возмущений, состояний, δ_A – функция переходов

$$\delta_A : Q \times Y \times [Z] \rightarrow Q : \langle q, y, [z] \rangle \mapsto q',$$

причем $\delta_A(q, y, e) = \delta_A(q, y) = q'$, λ_A / Λ_A – функция выходов Мили/Мура

$$\lambda_A : Q \times Y \times [Z] \rightarrow U : \langle q, y, [z] \rangle \mapsto u / \Lambda_A : Q \rightarrow U : q \mapsto u$$

с тем, что $\lambda_A(q, y, e) = \lambda_A(q, y) = u$ и q_H – начальное состояние.

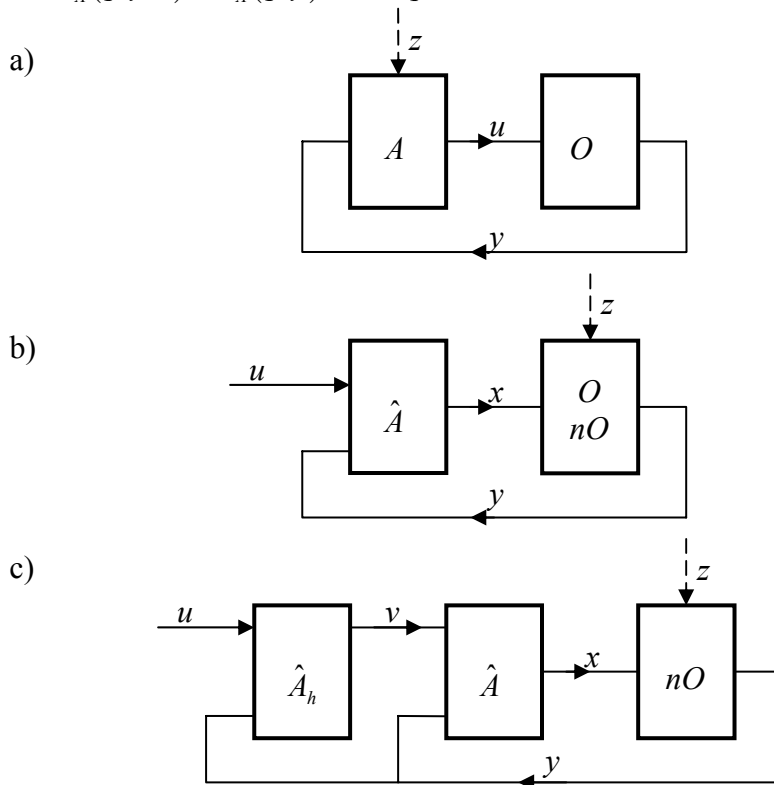


Рис. 3.1. Система автоматического логического управления а) по Глушкову, б) нововведенная, в) иерархическая.

Видимо, динамический автомат A (5) это наследство закона необходимого разнообразия [19], гласящего, что успешно справиться с разнообразием в объекте управления может только такой управляющий автомат, который сам обладает достаточным разнообразием.

Моделью системы управления (рис.3.1.а)), служит конечный автомат

$$N(A, O) = \langle [Z], S \times Q, \delta_L, \langle s_H, q_H \rangle \rangle, \quad (6)$$

где отношение, в частности функция, переходов δ_L моделирует взаимодействие O (4) и A (5) в $N(A, O)$ (6)

$$\delta_L : (S \times Q) \times (S \times Q) : \langle s, q, s', q' \rangle (= \langle s, q, \text{proj}_3 \delta_O(s, u, s'), \delta_A(q, y) \rangle),$$

где proj_3 – проекция на 3 – ю ось т.е. либо для управляющего автомата Мили

$$\delta_L(s, q, \text{proj}_3 \delta_O(s, \lambda_A(q, \Lambda_O(s)), s'), \delta_A(q, \Lambda_O(s))), \quad (7)$$

либо для управляющего автомата Мура

$$\delta_L(s, q, \text{proj}_3 \delta_O(s, \Lambda_A(q), s'), \delta_A(q, \Lambda_O(s))),$$

в частности

$$\delta_L : S \times Q \times Z \rightarrow S \times Q : \langle s, q, z \rangle \mapsto \langle s', q' \rangle (= \langle \text{proj}_3 \delta_O(s, u, s'), \delta_A(q, y, z) \rangle), \quad (7')$$

т.е. или для управляющего автомата Мили

$$\delta_L(s, q, z) = \langle \text{proj}_3 \delta_O(s, \lambda_A(q, \Lambda_O(s), z), s'), \delta_A(q, \Lambda_O(s), z) \rangle, \quad (8)$$

или для управляющего автомата Мура

$$\delta_L(s, q) = \langle \text{proj}_3 \delta_O(s, \Lambda_A(q), s'), \delta_A(q, \Lambda_O(s), z) \rangle.$$

Может удивить, что, хотя O (5) недетерминированный, δ_L (7') – функция, а не лишь отношение. Но, естественно предпологать, что воздействует ли явное возмущение z на A (9), то воздействует также на O (5), т.е. как бы $\text{proj}_3 \delta_O(s, u, s') = \delta_O(s, u, z)$.

Принципиально ту-же самую концепцию управления по Глушкову можно встретить в [1, 9]. К стати, можно придумать другое понимание?

Ограничимся конечно-автоматной моделью управляющего автомата вида Мили

$$A = \langle Y \times [Z], Q, U, \delta_A, \lambda_A, q_H \rangle. \quad (9)$$

Пример 3.1.: Пусть моделью ненаблюдаемого объекта управления есть диаграмма переходов (рис.3.2.а)) и требуется построить такой управляющий автомат [1], чтобы при произвольном начальном состоянии объекта автомат переходил бы в состояние q_3 и далее неограниченно долго в нем оставался. Возможная диаграмма переходов управляющего автомата приведена на рис.3.2. б). ■

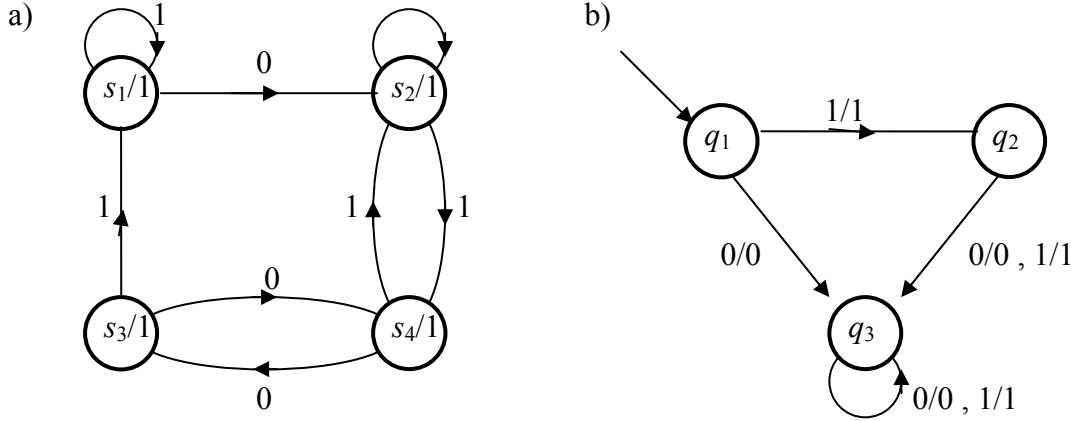


Рис. 3. 2. Диаграмма переходов: а) объекта, б) автомата из примера 3.1.

Названия конечно-автоматных моделей в примере 3.1., конечно, лишь условны; речь идет, скорее всего, о взаимодействии двух детерминированных автоматов в цепи обратной связи, причем вербально заданный автомат („управляющий“) ищется, нежели о систему управления.

Т е о р е м а 3. 1. Модель A (9) типа Мили сводится после минимизации числа ее состояний ($|Q| = 1$) к т.н. автомату без памяти

$$A = \langle Y \times [Z], X, \lambda_A \rangle, \quad (10)$$

где λ_A – функция выходов

$$\lambda_A : Y \times [Z] \rightarrow X : \langle y, [z] \rangle \mapsto x.$$

с тем, что $\lambda_A(y, e) = \lambda_A(y) = x$.

Д о к а з а т е л ь с т в о : Пусть $Q = \{q\}$; тогда (см. приложение 2) можно формально (но не фактически) игнорировать функцию переходов $\delta_A : \{q\} \times Y \times [Z] \rightarrow \{q\}$ и формально (но не фактически) модифицировать функцию выходов $\lambda_A : \{q\} \times Y \times [Z] \rightarrow X$ в форму $\lambda_A : Y \times [Z] \rightarrow X : \langle y, [z] \rangle \mapsto x$. Отношение, в частности функция, переходов δ_L системы управления примет в таком случае вид

$$\delta_L : ((S \times \{q\}) \times (S \times \{q\})) = S \times S : \langle s, s' \rangle (= \langle s, \text{proj}_3 \delta_o(s, \lambda_A(\Lambda_o(s)), s') \rangle) \quad (11)$$

в частности

$$\delta_L : (S \times \{q\} \times Z \rightarrow S \times \{q\}) = S \times Z \rightarrow S : \langle s, z \rangle \mapsto s' (= \text{proj}_3 \delta_o(s, \lambda_A(\Lambda_o(s), z), s')). \quad (12)$$

Сравнивая соответствия (7) и (11), в частности (8) и (12), убеждаемся, что обе формы отношения, спец. функции, δ_L полностью описывают взаимодействие O (4) и A (9), (10) в системе управления (см. рис.3.3.) с тем, что составляющая $\delta_A(q, \Lambda_o(s))$ в (7), в частности $\delta_A(q, \Lambda_o(s), z)$ в (8), оказывается избыточной. \square

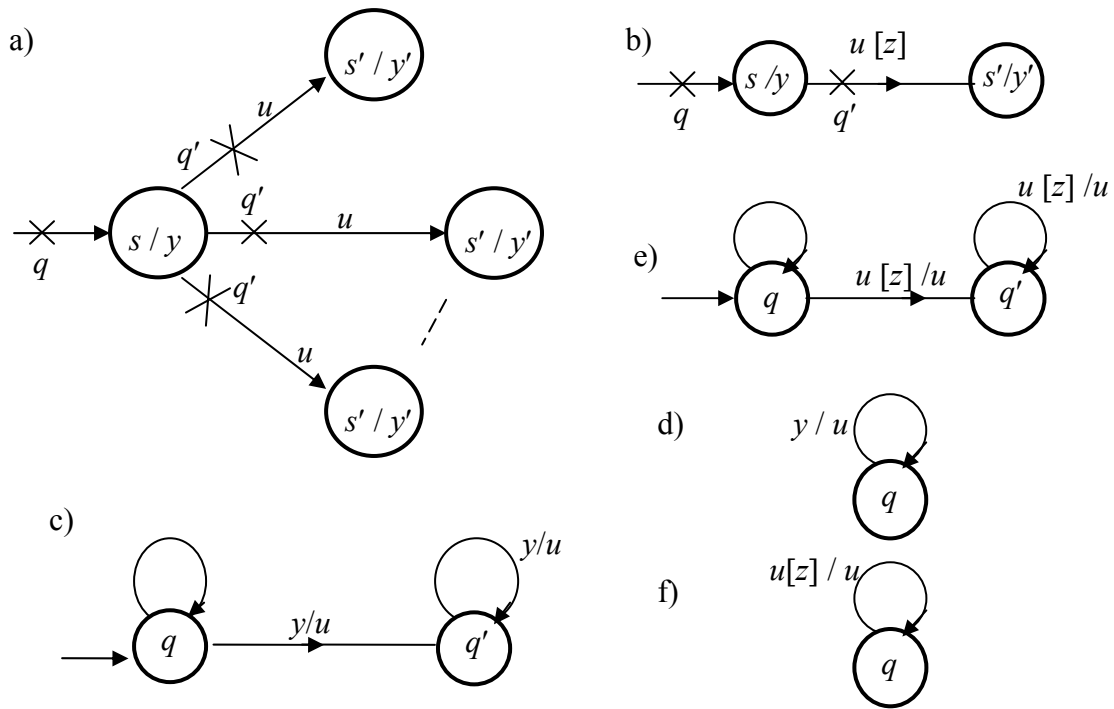


Рис. 3.3. Фрагменты диаграмм переходов а) недетерминированного, б) детерминированного O ; с), е) A ; д), ф) минимального управляющего автомата, описывающие взаимодействие O и A в системе управления, где крестик \times отмечает состояния A .

Само собой разумеется, что отдадим предпочтение минимальным управляющим автоматам Мили с одним состоянием.)*

Пример 3.2. : Построить динамический автомат A (таб.3.1.с)) и его минимальную статическую форму \hat{A} (таб.3.1.d)), управляют которые лифтом трехэтажного дома, вырабатывая управление: подъем, спуск и stop. Случайный выбор этажа (измеримое возмущение) осуществляется кнопками z_1, z_2 и z_3 . Достижение лифтом i -го этажа ($i = 1, 2, 3$) отмечается сигнализатором положения – состояния – лифта. Функция переходов системы $N(A, \text{лифт})$ или $N(\hat{A}, \text{лифт})$ видна в таб.3.1.a) с обозначенными собственными траекториями состояний или в таб.3.1.b), содержащей надлежащее управление. Диаграмма переходов системы $N(A, \text{лифт})$ с крестиками, отмечающими состояния автомата A , имеется на рис.3.4. По таб.3.1с) можно, с одной стороны, просто усмотреть недетерминизм автоматов A и \hat{A} , фиктивно не располагая возмущающими входами z_1, z_2 и z_3 , и с другой, показать, что $|Q_{\text{лифта}}| = 3 |S_{\text{лифта}}| - 2$. В [17], откуда пример заимствован, считается, что управление лифтом осуществляется кнопками (без памяти) z_1, z_2 и z_3 и непосредственно предлагается диаграмма переходов управляющего автомата, которая после того упрощается. ■

)* Объекты с одним состоянием суть статические и являются частным случаем объектов динамических. Не придерживаемся традиционной классификации объектов по их поведению, ибо хотя статические объекты ведут себя всегда комбинационно, объекты динамические могут быть комбинационными или последовательностными.

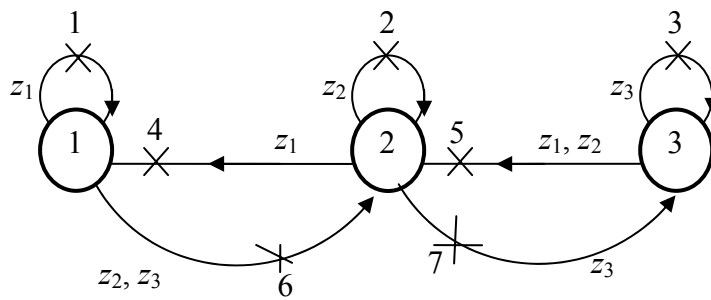


Рис. 3. 4. Диаграмма переходов системы $N(A, \text{лифт})$ из примера 3.2.

Таб. 3. 1.

a)

$s \backslash z$	s'		
	z_1	z_2	z_3
1	↑ (1)	↓ 2	↓ 2
2	1 ↑	(2) ↑	↓ 3
3	2 ↑	2 ↑	(3) ↓

b)

$s \backslash zu$	s'								
	z_1 stop	z_1 спуск	z_1 подъем	z_2 stop	z_2 спуск	z_2 подъем	z_3 stop	z_3 спуск	z_3 подъем
1	(1)	-	-	-	-	2	-	-	2
2	-	1	-	(2)	-	-	-	-	3
3	-	2	-	-	2	-	(3)	-	-

c)

$q \backslash zs$	q'/u								
	$z_1 1$	$z_1 2$	$z_1 3$	$z_2 1$	$z_2 2$	$z_2 3$	$z_3 1$	$z_3 2$	$z_3 3$
1	(1)/stop			6/под.			6/под.		
2		4/спуск			(2)/stop			7/под.	
3			5/спуск			5/спуск			(3)/stop
4	1/stop	(4)/спуск							
5			(5)/спуск		2/stop	(5)/спуск			
6				(6)/под.	2/stop		(6)/под.		
7								(7)/под.	3/stop

d)

z	z_1	z_1	z_1	z_2	z_2	z_2	z_3	z_3	z_3
s	1	2	3	1	2	3	1	2	3
u	stop	спуск	спуск	подъем	stop	спуск	подъем	подъем	stop

Пример 3.3.: Пусть задана технологическая аппаратура (рис.3.4.) – $N (A, r_1, r_2, p)$, включающая задвижки K_1, K_2 , причем их положение управляется командами закрыть – \bar{u}_1 или открыть $u_i (i = 1,2)$ с тем, что положение истоковых задвижек не контролируется. По каждому резервуару контролируется нижний – h_i^{\min} и верхний – h_i^{\max} уровень жидкости в соответствующем баке (см. таб.3 а)). Насос p управляется по команде \bar{p} – выключен или p – включен, причем насос никогда не работает в холостую. Предположим что количество впускаемой жидкости равно истоковому количеству в случае одновременного наполнения обоих резервуаров; наполняется ли лишь один бак, то количество впускаемой жидкости выше истокового. Очевидно, что $p = u_1 \vee u_2$. В таб. 3.2.b) находится отношение переходов $N (A, r_1, r_2, p)$ и в таб.3.2.c) имеются собственные траектории состояний. Построить минимальный управляющий автомат – таб.3.2.d); но разработанный автомат оказывается недетерминированным. Решение проблемы приведенного недетерминизма имеется в примере 3.7. ■

Таб. 3. 2.

a)

h_i^{\min}	h_i^{\max}	резервуар i	состояние	u_i
0	1	полный	2	закрыть
0	0	полупустой	1	–
1	0	пустой	0	открыть

b)

$u_1 u_2$		$s_1 s_2 \backslash p$		$s'_1 s'_2$	
				p	\bar{p}
1 1	0 0	(00), 01, 10,	(00)		
1 1	0 1	(01), 11, 02,	(01), 00		
1 0	0 2	(02), 12	(02), 01		
1 1	1 0	(10), 11, 20	(10), 00		
1 1	1 1	(11), 12, 21	(11), 01, 10		
1 0	1 2	(12), 22	(12), 11, 02		
0 1	2 0	(20), 21	(20), 10		
0 1	2 1	(21), 22	(21), 11, 20		
0 0	2 2	–	(22), 12, 21		

c)

p	p			\bar{p}
$u_1 u_2$	11	10	01	00
$s_1 s_2$	00,01, 10,11	02 12	20 21	00,01,02,10,11,12,20,21,22

d)

p	p			\bar{p}
$s_1 s_2$	00,01, 10,11,	02 12	20 21	00,01,02,10,11,12,20,21,22
$u_1 u_2$	11	10	01	00

e)

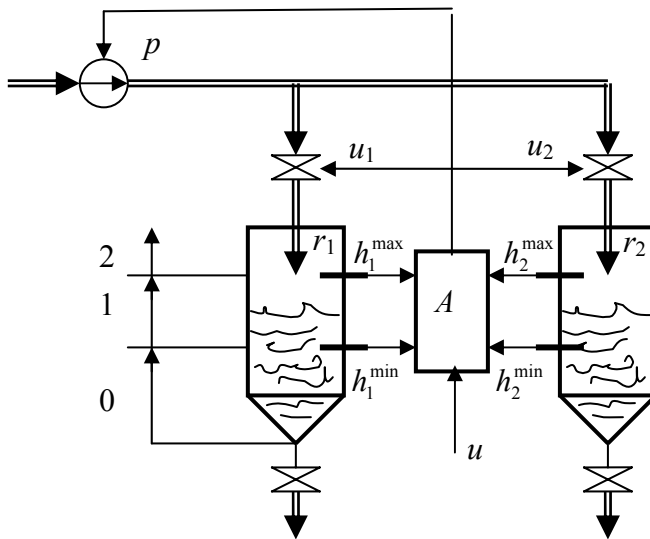
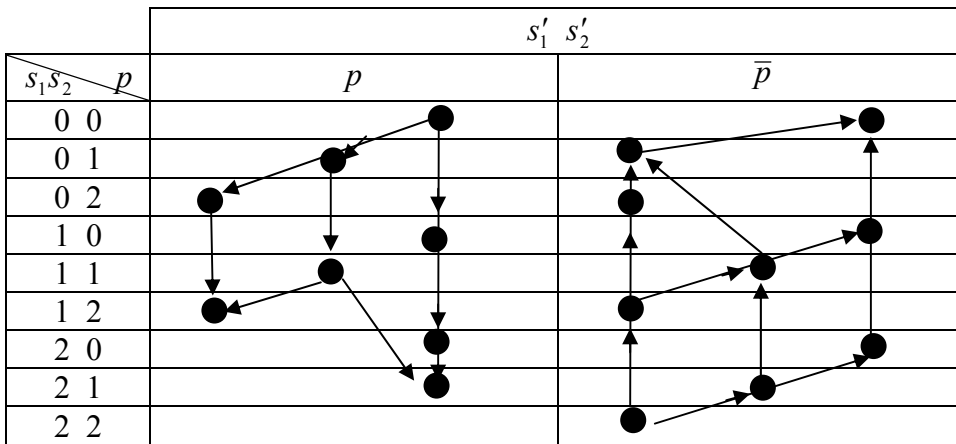


Рис. 3. 4. Система $N(A, r_1, r_2, p)$ из примера 3.3.

Пример 3.4. Построить управляющий автомат (таб.3.3.a,b)) по заданной граф-схеме [9] поведения объекта управления (рис.3.5.), где крестиками отмечены состояния управляющего автомата. Реконструированная конечно-автоматная модель объекта управления, считая ее наблюдаемой, имеется в таб.3.3.c) и эволюционирует по таб.3.3.d). ■

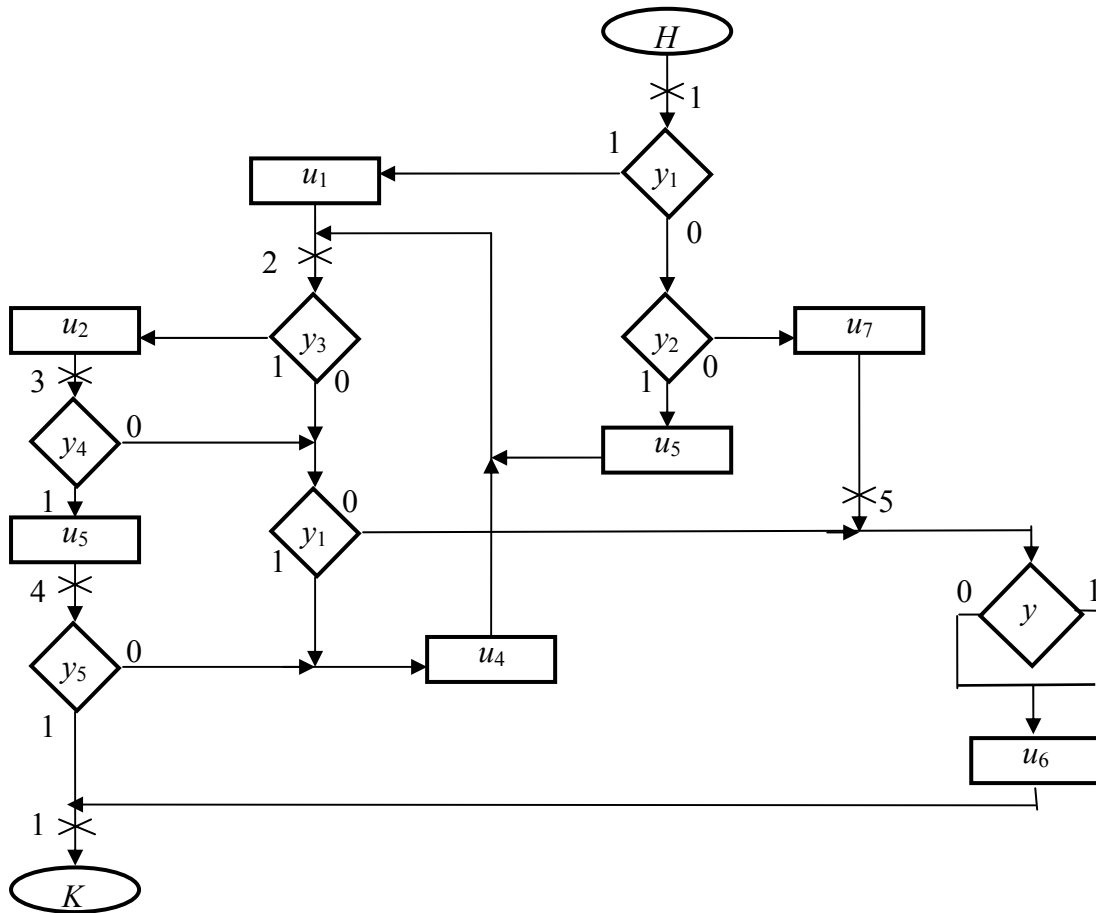


Рис. 3. 5. Отмеченная граф-схема объекта управления из примера 3. 4.

Таб. 3. 3.

a)

		q' / u										
$q \setminus y$	y_1	$\bar{y}_1 y_2$	$\bar{y}_1 \bar{y}_2$	y_3	$y_1 \bar{y}_3$	$\bar{y}_1 \bar{y}_3$	y_4	$y_1 \bar{y}_4$	$\bar{y}_1 \bar{y}_4$	y_5	\bar{y}_5	1
1	2 / u_1	2 / u_5	5 / u_7									
2				3 / u_2	2 / u_4	1 / u_6						
3							4 / u_3	2 / u_4	1 / u_6			
4										1 / u	2 / u_4	
5												1 / u_6

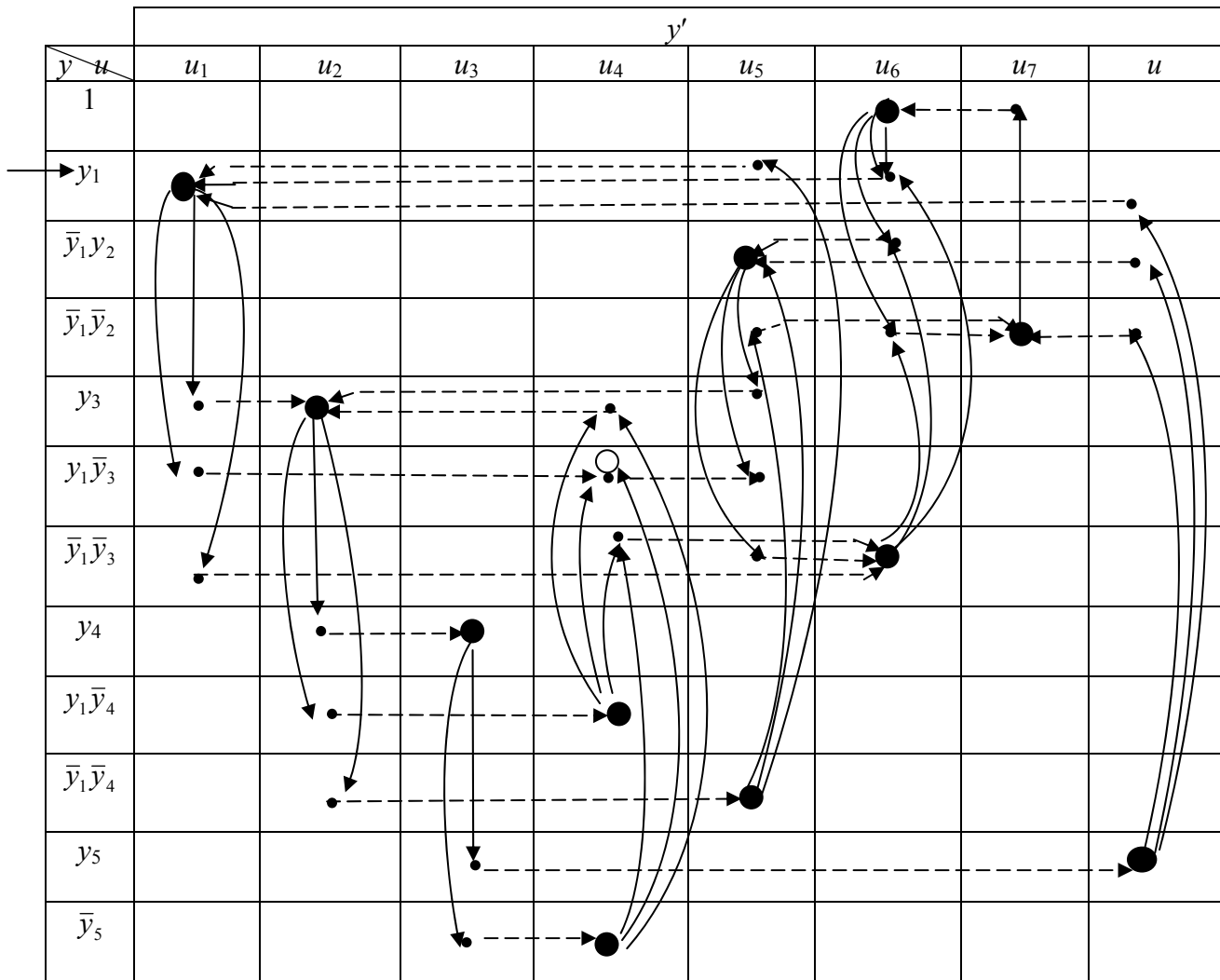
b)

y	y_1	$\bar{y}_1 y_2$	$\bar{y}_1 \bar{y}_2$	y_3	$y_1 \bar{y}_3$	$\bar{y}_1 \bar{y}_3$	y_4	$y_1 \bar{y}_4$	$\bar{y}_1 \bar{y}_4$	y_5	\bar{y}_5	1
u	u_1	u_5	u_7	u_2	u_4	u_6	u_3	u_4	u_6	u	u_4	u_6

c)

		y'						
$y \setminus u$	u_1	u_2	u_3	u_4	u_5	u_6	u_7	u
1						$y_1, \bar{y}_1 y_2,$ $\bar{y}_1 \bar{y}_2$		
y_1	y_3, y_1 $\bar{y}_3, \bar{y}_1, \bar{y}_3$							
$\bar{y}_1 y_2$					$y_3, y_1 \bar{y}_3,$ $\bar{y}_1 \bar{y}_3$			
$\bar{y}_1 \bar{y}_2$							1	
y_3		$y_4, y_1 \bar{y}_4,$ $\bar{y}_1 \bar{y}_4$						
$y_1 \bar{y}_3$				$y_3, (y_1 \bar{y}_3),$ $\bar{y}_1 \bar{y}_3$				
$\bar{y}_1 \bar{y}_3$						$y_1, \bar{y}_1 y_2,$ $\bar{y}_1 y_2$		
y_4			y_5, \bar{y}_5					
$y_1 \bar{y}_4$				$y_3, y_1 \bar{y}_3,$ $\bar{y}_1 \bar{y}_3$				
$\bar{y}_1 \bar{y}_4$						$y_1, \bar{y}_1 y_2,$ $\bar{y}_1 \bar{y}_2$		
y_5								$y_1, \bar{y}_1 y_2,$ $\bar{y}_1 \bar{y}_2$
\bar{y}_5				$y_3, y_1 \bar{y}_3,$ $\bar{y}_1 \bar{y}_3$				

d)



Пример 3. 5. Построить «управляющее устройство» светофора, фонари которого управляют транспортным движением на магистрали (m) и на перекрещивающей её локальной дороге (l) [23]. Пешеходы или водители, желающие пересечь магистраль, заявляют об этом сигналом stop. Обозначим зажженный красный, желтый и зелёный фонарь, направленный на магистраль или на локальную дорогу, соответственно через $K_m, Ж_m, З_m$ или $K_l, Ж_l, З_l$ и состояния проектируемой цепи через упорядоченные пары $K_m Ж_l, З_m K_l, Ж_m K_l, K_m З_l$. Считая, что при подаче сигнала stop, находится цепь в состоянии $З_m K_l$ и $K_m З_l$ соответственно в течение τ_m и τ_l времени, получают диаграмму переходов не «управляющего устройства» светофора, а светофора, в состав которого входят динамическая цепь и фонари, подключенные к её выходам. Диаграмма переходов цепи имеется на рис.3.5. ■

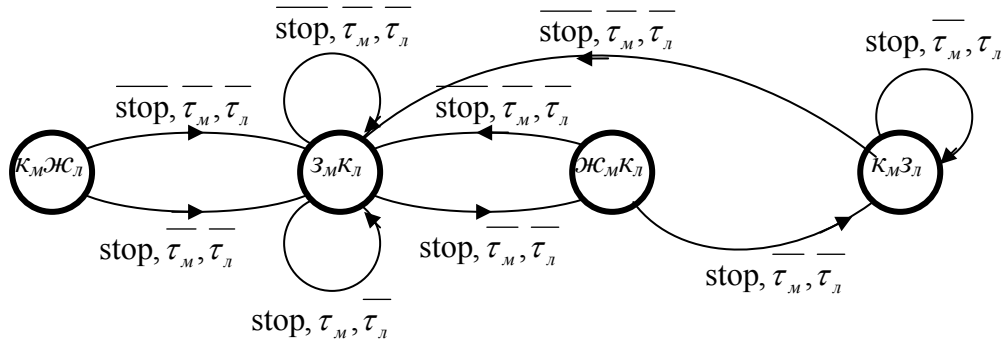
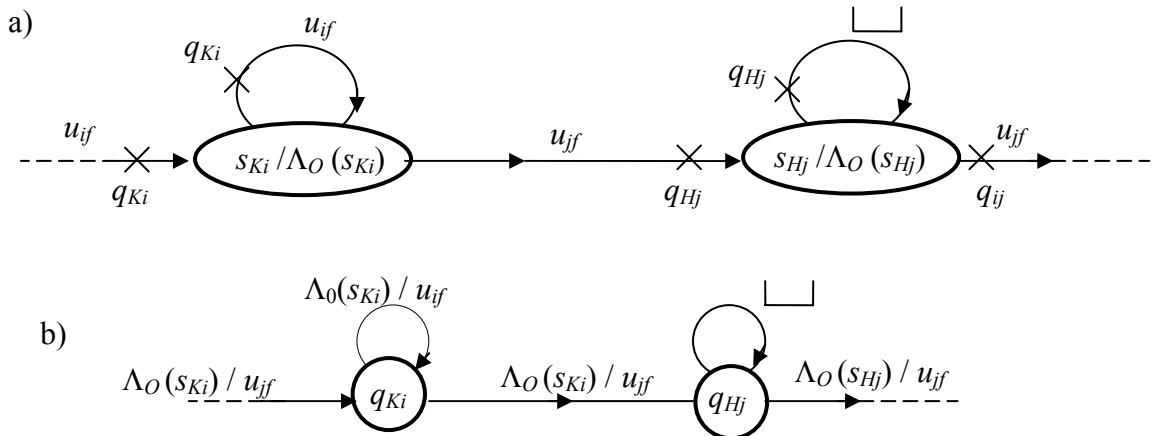


Рис. 3. 5. Диаграмма переходов светофора из примера 3. 5.

Теорема 3.2. Традиционной концепции логического управления или с динамическим, или со статическим автоматом управления A (5) свойственен недетерминизм, маскируется который с одной стороны измеримыми возмущениями, и с другой задающим управлением субъекта.

Доказательство: Воспользовавшись требованием, накладываемым на собственные траектории состояний, рассмотрим переход из одной собственной траектории состояний T_i в другую T_j объекта, т.е. рассмотрим переход из конечного состояния $s_{Ki} (\delta_0(s_{Ki}, u_{if}, s_{Ki}), \text{ спец. } \delta_0(s_{Ki}, u_{if}) = s_{Ki})$ в начальное состояние $s_{Hj} (\delta_0(s_{Hj}, \square, s_{Hj}), \text{ спец. } \delta_0(s_{Hj}, \square) = s_{Hj})$ под воздействием „автоматного“ управления u_{if} (рис.3.6.a)). Очевидно, что $u_{if} = \lambda_A(q_{Ki}, \Lambda_O(s_{Ki})) = \lambda_A(\Lambda_O(s_{Ki})) = u_{if} = u$, (рис.3.6.b,c)) хотя настоятельно требуется, чтобы $u_{if} \neq u_{jf}$. Поэтому введем, с одной стороны, измеримые возмущения (располагаем ли мы ими) $z_i, z_j (z_i, z_j \in Z; z_i \neq z_j \text{ при } i \neq j)$ и получим $u_{if} = \lambda_A(q_{Ki}, \Lambda_O(s_{Ki}), z_i) = \lambda_A(\Lambda_O(s_{Ki}), z_i) \neq \lambda_A(q_{Kj}, \Lambda_O(s_{Kj}), z_j) = \lambda_A(\Lambda_O(s_{Kj}), z_j) = u_{jf}$, т.е. $\delta_0(s_{Ki}, u_{if}, z_i) = s_{Ki}$ и $\delta_0(s_{Kj}, u_{jf}, z_j) = s_{Hj}$, и с другой, задающее управление $u_i = u_{if}$ и $u_j = u_{jf}$ ($u_i \neq u_j$ при $i \neq j$), выбирает которым субъект траекторию либо T_i , либо T_j . Автор надеется, что одинаковое обозначение (которого нельзя пока избежать) управления, выдаваемого управляющим автоматом („автоматное“ управление), и управления, задаваемого субъектом (задающее управление), не вызовет смущение. ■



с)

$$\dots, \Lambda_O(s_{Ki}) / u_{if}, \Lambda_O(s_{Ki}) / u_{if}, \Lambda_O(s_{Ki}) / u_{if}$$

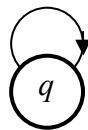


Рис. 3. 6. Взаимодействие: а) объекта управления и управляющего автомата: б) динамического, с) статического.

Пример 3. 6. Пусть задан объект управления (таб.3.4.а) с тем, что состояние $s_H = 1$ должно быть управляемо в состояние $s_K = 2$. Построить минимальный управляющий автомат, располагая или только таб.3.4.а) и таб.3.4.с), или измеримыми возмущениями, т.е. таб.3.4.б) и таб.3.4.д), или задающими управлениями – таб.3.4.е). Сигнал u в таб.3.4. а), б), с), д) – „автоматное“, а u в таб.3.4.а), е) – задающее управление; x в таб.3.4.е) – машинное управление (см. раздел 4.).

Таб. 3. 4.

а)

$s \backslash u$		s'	
		a	b
→	1	↓ 2	–
←	2	↓ 3	(2) ↑
	3	↓ 4 ↓	↑ 2 ↑
	4	↓ 5 ↓	↑ 3 ↑
	5	(5) ↓	4 ↑

б)

$s \backslash uz$		s'	
		az_1	az_2
↔	1	↓ 2	–
	2	↓ 3 ↓	(2) ↑
	3	↓ 4 ↓	↑ 2 ↑
	4	↓ 5 ↓	↑ 3 ↑
	5	(5) ↓	4 ↑

с)

s	1	2	3	4	5
u	a	a, b	a, b	a, b	a, b

д)

z	z_1					z_2			
s	1	2	3	4	5	2	3	4	5
u	a					a			

е)

u	a					b			
s	1	2	3	4	5	2	3	4	5
x	α	β	γ	δ	ε	β	γ	δ	ε

Пример 3. 7.: Вернемся к примеру 3.3.. Введем вдобавок задающее управление $u \in \{\text{start}, \text{stop}\}$ в управляющий автомат и модифицированное отношение переходов установки примет вид по таб.3.5.а). Отсюда функция выходов расширенного управляющего автомата – таб. 3.5.б).

Таб. 3. 5.

а)

$s_1 s_2 \backslash up$		$s'_1 s'_2$		
		start p	start \bar{p}	stop \bar{p}
0	0	(00), 01, 10	–	(00)
0	1	(01), 11, 02	–	(01), 00
0	2	(02), 12	–	(02), 01
1	0	(10), 11, 20	–	(10), 00
1	1	(11), 12, 21	(11), 01, 10	(11), 01, 10
1	2	(12), 22	(12), 11, 02	(12), 11, 02
2	0	(20), 21	–	(20), 10
2	1	(21), 22	(21), 11, 20	(21), 11, 20
2	2	–	(22), 12, 21	(22), 12, 21

б)

(p)	p			\bar{p}				\bar{p}
u	start			start				stop
$s_1 s_2$	00, 01 10, 11	01 12	20 21	11	12	21	22	00, 01, 02, 10, 11 12, 20, 21, 22
$u_1 u_2$	11	10	01	11	10	01	00	00

4. Комментарий к традиционному управлению по Глушкову

Пусть имеется динамический объект, собственные траектории состояний которого устраивают заказчика. В таком случае не зачем управлять заданным объектом в системе автоматического логического управления с обратной связью, а хватит лишь объект управления завести, ожидая достижение конечного состояния объектом вдоль субъектом избранной траектории, выполняющейся автоматически в объекте. Примером автоматического выполнения собственной траектории состояний может служить стрельба заряженного и подготовленного к стрельбе стрелкового оружия АК – 47 после нажатия крючка. Вот почему, на взгляд автора, отождествляется проектирование динамических логических цепей (структурных моделей) и логическое управление – см. пример 3.5.

Зачастую считают динамическими объектами подавляющее большинство объектов, которые, в самом деле, динамическими не являются. Займемся поэтому идентификацией водяного бака из рис.4.1. Строя отношение переходов резервуара, связывают с баком, по умолчанию, положение уровня воды относительно датчиков D_1 и D_2 , причем каузальная связь между уровнем воды и открытием задвижки существует пока только в воображении идентифицирующего бак субъекта. Таким образом, в состав бака входит, неумышленно и само лицо, идентифицирующее бак, отождествляет которое самого себя с концепционным управляющим автоматом. Наблюдателя, разумеется, нельзя ни в коем случае упрекать в искаженном видении бака. Ведь субъект, только мыслительно управляя баком, может прийти к модели – собственно чего? Резервуар сам по себе явно не целостный и, стало быть, лишь **потенциально – динамический**.

Итак, очевидно, что собственные траектории состояний не выполняются в технологическом логическом потенциально-динамическом объекте, спонтанно, а ими надо управлять.

Возможные возражения читателя против сказанного можно опровергнуть, ибо технологи до того, пока не привлекут разработчиков управляющей автоматики, проектируют объекты

так, что в их состав вводят всякого рода движители, как например ходовые винты; тяговые, цепные, ременные, зубчатые и гидромеханические передачи; подающие механизмы, сбрасыватели, электро-передачи, продуктопроводы и т.п., обеспечивающие, по сути дела, кажущееся стихийное выполнение траекторий в объекте. Хотя движители естественно движутся, в системе автоматического логического управления, как ни странно, играют роль статических элементов. Но, упомянутые управляющие движители не считают, почему-то,

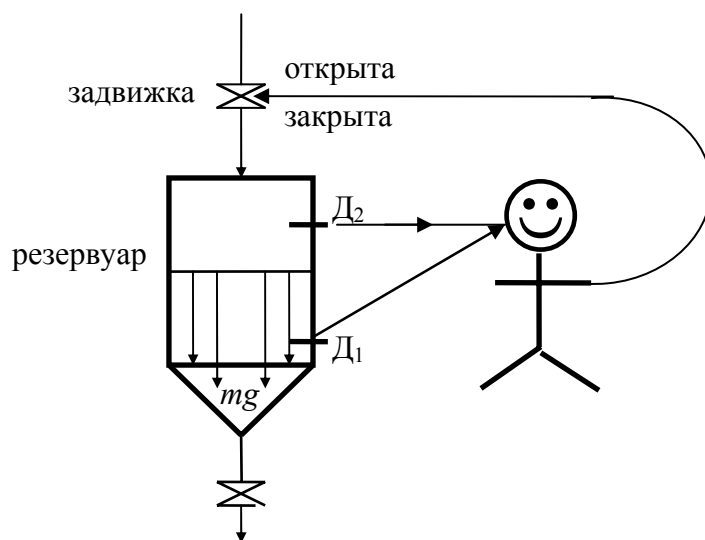


Рис. 4. 1. Земное тяготение (mg), субъект и потенциально-динамический бак как доли целостного динамического агрегата.

достаточно важными, чтобы войти в состав управляющего автомата. И бывает, что разработчику автомата заданного объекта, в самом деле, нечего делать, ибо объект „самоуправляем“, т.е. объект уже не потенциально-динамический, а в месте с движителями образует динамическую систему управления. Ярким примером выше сказанного может служить стрелковое оружие – автомат АК – 47. Перед выстрелом заводится замок, патрон вводится и возвращая пружиной сжатия затвора замок, патрон попадает в патронник. Нажимая спусковой крючок, подводят патрон к выстрелу. Часть газов, возникших выстрелом, отводится к обратному поршню, повторно, вместе с пружиной сжатия, заводящему затвор и, таким образом, стрельба автоматически продолжается. Применив холостой патрон и не используя ствольную насадку, автомат станет не управляемой винтовкой. Короче говоря, АК – 47 без ствольной насадки и с холостыми патронами не является динамическим логическим объектом.

Стало быть, проектировщик автоматики заданного объекта, считает объект всегда динамическим или в своем воображении, или игнорируя управляющие движители, внедренные в объект технологом.

В случае объекта виднее чем в случае логических цепей с обратными связями, что переход между состояниями объекта не вызывает одно лишь управляющее воздействие, возможно вместе с возмущением. Внешнее воздействие на систему управления переход в системе лишь инициирует, а исполняет переход его исходное состояние. Например, вряд ли можно утверждать, что нажатие спускового крючка АК – 47 вызывает автоматически повторяемые выстрелы, но можно согласиться, что нажатие спускового крючка разрешает стрельбу, а сам выстрел автомата приводит к последовательному выстрелу. Или, сигнал дежурного по станции инициирует отправление поезда, находящегося в станции, но едва ли сигнал отправления его исполняет, ибо только движение двигателя перемещает поезд

из остановки в остановку. Или, в случае лягушки, прыгающей из одного листа кувшинки на другой, вряд ли один стимул может вызвать прыжок лягушки; стимул прыжок лишь инициирует, а исполняет прыжок сама лягушка. Приведенный факт иллюстрируют примеры 3.2. и 3.3., где аппаратура, состоящая из насоса и двух резервуаров и кабина лифта представляют собой лишь потенциально-динамические объекты, которые только совместно с управляющими автоматами образуют динамическую систему автоматического управления. Метафорически сказано: управляющий автомат „оживляет“ „мертвый“ потенциально-динамический объект управления. Следует однако отметить, что потенциально-динамический объект не есть статический.

Способы построения конечно-автоматных моделей и наложения их на объекты управления в теорию логического управления обычно не входят; анализ этих способов важная нерешенная проблема. Факт, что хотя конечные автоматы объектов управления приходится разрабатывать по интуиции, не дает основание ими, как правило, не пользоваться. Тем более не уместно игнорировать конечные автоматы с возмущающими входами, хотя нередко на объекты измеримые возмущения воздействуют, а наоборот, не воздействуют ли измеримые возмущения на объект управления, воздействуют иногда на управляющий автомат [6,14,21]. Более того, хотя и упоминаются фундаментальные понятия как наблюдаемость и управляемость объектом [17], считают, хотя объект управления, как правило, задается лишь вербально и зачастую не на состоятельном уровне детализации (см. пример 3.4.), по умолчанию, объект управления, как будто само собой разумеется, наблюдаемым и управляемым [4,5,8,11,12,13,17]. Главное внимание уделяется построению конечно-автоматной модели управляющего автомата [4,5,7,8,9,11,12,13,15,17,23], который представляет собой, таким образом, некоторый „суррогат“ объекта управления, не располагая моделью объекта.

Стараясь управлять недетерминированным динамическим объектом, т.е. заставляя объект автоматически двигаться вдоль принужденной траектории состояний, ибо собственные траектории состояний в объекте не устраивают технолога, и не располагая измеримыми возмущениями или задающим управлением или игнорируя требование, накладываемое на собственные траектории состояний, вряд ли можно обратиться к системе управления по Глушкову.

Статический, в отличие от динамического, управляющий автомат не располагает „памятью“ для того, чтобы, в случае импульсных датчиков состояний объекта управления, проложить мост над продолжительностью перехода между состояниями в объекте управления. Пусть состояния двоичного объекта кодируются унитарно $(k: S \rightarrow \{0,1\}^{|S|} : s_i \mapsto 0_1 0_2 \dots 0_{i-1} 1_i 0_{i+1} \dots 0_{|S|}, \text{ т.е. } w_H(k(s_i)) = 1, \text{ где } w_H - \text{вес Хемминга})$. Тогда, если автомат статический и если датчики состояний в объекте потенциальные (уровневые) D_y или импульсные D_n , то для, без потери общности, достоверного перехода $s' = \delta_o(s, x, [z])$ имеет место рис.4.2., где элементы запрета или задержек препятствуют состязаниям в кодовых словах состояний s и s' , а D_y или опорный RS – триггер „отождествляют“ промежуточное состояние перехода из s в s' с исходным состоянием s перехода (см. приложение 3) [24,25].

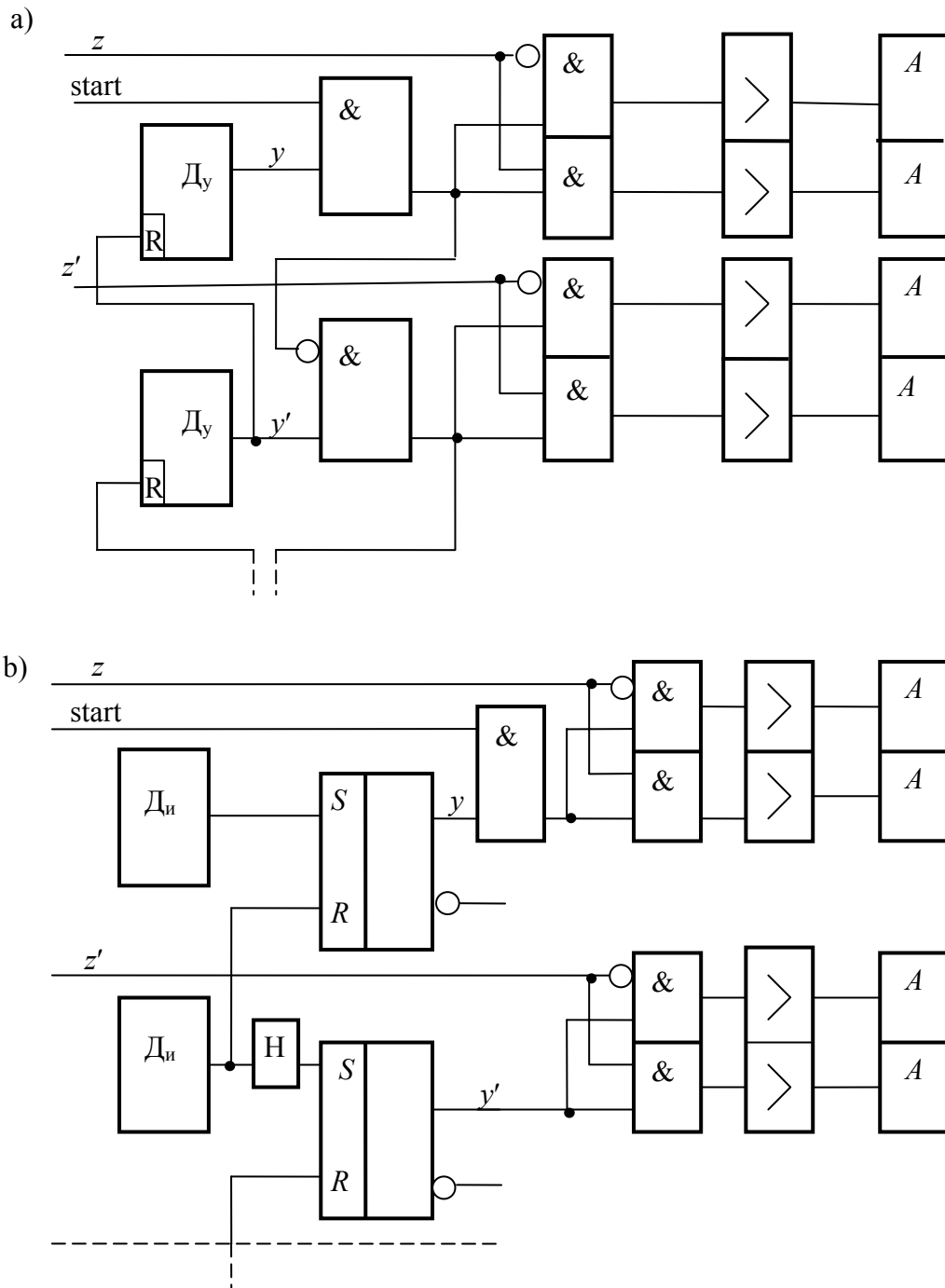


Рис.4. 2 Фрагменты цепочек состоящих из: а) потенциальных (D_y), б) импульсных (D_n) датчиков, совершающих переход $s' = \delta_o(s, x, [z])$ в двоичной системе управления, из усилителей, логических переходных устройств и исполнительных механизмов (акторов) A .

Займемся еще раз минимальным, по числу состояний, управляющим автоматом Мили. Интересно отметить, что автор, когда независимо от Беллмана показал, что управляющий автомат Мили статический [24], встретился с преобладающей отрицательной реакцией теоретиков, а не практиков. Но, защитившись авторитетом и назвав статический управляющий автомат Беллмановым [26], возражения утихли. Поскольку принцип

управления по Беллману: **управление динамическим объектом является функцией состояния объекта управления**, лишь постулируется, стоит его доказать. Действительно, поскольку исходное состояние любого перехода каждой принужденной траектории состояний в объекте управления является необходимой причиной перехода, то, подводя реакцию y ($y = \Lambda_0(s)$, или же $s = \Lambda_0^{-1}(y)$) на вход объекта управления, располагают так-же причиной достаточной перехода. Таким образом, состояние объекта управления становится одновременно необходимой и достаточной причиной перехода в объекте управления в системе управления – рис.3.1.а).

4. Нововведенное традиционное логическое управление динамическим объектом

Напомним, что если собственные траектории состояний (2)/(3) в объекте устраивают технолога, то хватит управлять объектом директивно.

Гораздо интереснее выглядит случай, когда, хотя располагаем динамическим объектом O (1), его собственные траектории состояний не устраивают заказчика. Пусть поэтому технолог настаивает на выполнении собственных траекторий состояний, избираемых задающим управлением, устраивают его которые в задуманном конечном автомате (прототипе)

$$T = \langle U \times [Z], Q, V, \delta_T, \Lambda_T, q_H \rangle, \quad (13)$$

где U, Q, V – алфавит соответственно задающих управлений для выбора требуемых собственных траекторий состояний в системе управления, состояний, выходов, δ_T – отношение

$$\delta_T : Q \times U \times Q : \langle q, u, q' \rangle,$$

в частности функция

$$\delta_T : Q \times U \times [Z] \rightarrow Q : \langle q, u, [z] \rangle \mapsto q',$$

переходов, Λ_T – функция (инъективная)

$$\Lambda_T : Q \rightarrow V : q \mapsto v$$

выходов и q_H – начальное состояние.

Рассмотрим систему управления из рис.3.1.б). Моделью статического управляющего автомата есть т.н. автомат без памяти

$$\hat{A} = \langle U \times Y \times [Z], X, \hat{\lambda}_A \rangle, \quad (14)$$

где X – алфавит машинного управления $\hat{\lambda}_A$ – функция выходов (машинных управлений)

$$\hat{\lambda}_A : U \times Y \times [Z] \rightarrow X : \langle u, y, [z] \rangle \mapsto x.$$

Располагаем, надеясь, что наблюдаемым и управляемым, динамическим объектом

$$O = \langle X \times [Z], S, Y, \delta_O, \Lambda_O, s_H \rangle, \quad (15)$$

где S, Y – алфавит соответственно состояний, выходов, δ_O – отношение

$$\delta_O : S \times X \times S : \langle s, x, s' \rangle,$$

в частности функция

$$\delta_O : S \times X \times [Z] \rightarrow S : \langle s, x, [z] \rangle \mapsto s'$$

переходов, Λ_O – функция выходов

$$\Lambda_O : S \rightarrow Y : s \mapsto y$$

и s_H – начальное состояние. Моделью инновированной системы автоматического логического управления служит конечный автомат

$$N(\hat{A}, O) = \langle U \times [Z], S, \delta_L, s_H \rangle, \quad (16)$$

где δ_L – отношение, в частности функция, переходов

$$\delta_L : S \times U \times S : \langle s, u, s' \rangle \left(= \langle s, u, \text{proj}_3 \delta_O (s, \hat{\lambda}_A(u, \Lambda_O(s)), s') \rangle \right),$$

спец.

$$\delta_L : S \times U \times [Z] \rightarrow S : \langle s, u, [z] \rangle \mapsto s' \left(= \delta_O (s, \hat{\lambda}_A(u, \Lambda_O(s)), [z]) \right)$$

Пусть требуется в концепционном объекте (образце) T (13) выполнение некоторой собственной траектории, причем располагаем объектом O (15), в котором (после кодирования состояний) желаемая траектория представляет собой не собственную, а некоторую траекторию состояний принужденную.

Т е о р е м а 4. 1. Выше упомянутый имеющийся объект O (15) представляет собой возможного заменителя в декомпозиции концепционного объекта T (13) прототипа – рис. 3.1.b) [27,28] (см. приложение 4).

Д о к а з а т е л ь с т в о : Пусть заланы кодирования состояний $k : Q \rightarrow S : q \mapsto s$ и выходов $\kappa : V \rightarrow Y : v \mapsto y$. Требуется найти автомат $\hat{A} = \langle U \times Y \times [Z], X, \hat{\lambda}_A \rangle$, или же $\hat{\lambda}_A$, при существовании мономорфизма $\langle k, \kappa \rangle$ образца T (13) в систему управления $N(\hat{A}, O)$ (16). Иными словами, выполняется ли импликация

$$\delta_T(q, u, q') \Rightarrow \delta_L(k(q), u, k(q')) \Rightarrow \delta_O(k(q), \hat{\lambda}_A(u, \Lambda_O(k(q))), k(q')), \quad (17)$$

в частности равенство

$$\delta_T \circ k = k \circ \delta_L,$$

или же

$$k(\delta_T(q, u, [z])) = \delta_L(k(q), u, [z]) = \delta_O(k(q), \hat{\lambda}_A(u, \Lambda_O(k(q))), [z]), \quad (18)$$

и равенство

$$\Lambda_T \circ \kappa = k \circ \Lambda_O,$$

или же

$$\kappa(\Lambda_T(q)) = \Lambda_O(k(q)).$$

ибо диаграмма композиций функций (рис.4.1.) коммутативна, то находим машинное управление $x = \hat{\lambda}_A(u, \Lambda_O(k(q)), [z])$. \square

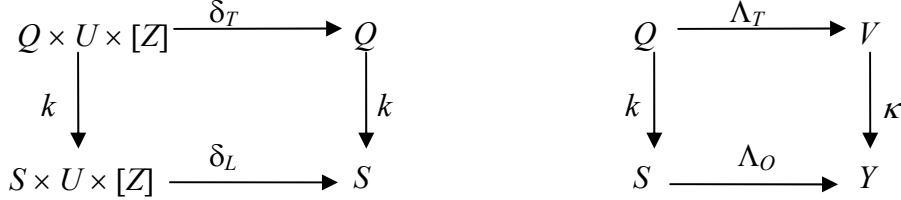


Рис.4.1. Диаграммы композиций $\delta_T \circ k$ и $k \circ \delta_L$ или $\Lambda_T \circ \kappa$ и $k \circ \Lambda_O$ функций $\delta_T, \delta_L, \Lambda_T, \Lambda_O, k, \kappa$.

О п р е д е л е н и е 4. 1. Удастся ли по импликации (17), спец. по равенству (18) определить функцию выходов управляющего автомата \hat{A} (14) при заданном кодировании состояний, то заданный объект является **управляемым**.

Сказано по другому, управляемый объект управления (заменитель желаемого автомата) способен посредством своих, машинным управлением принужденных траекторий состояний, имитировать требуемые собственные траектории состояний в конечном автомате, T (13).

О п р е д е л е н и е 4. 2. Заменитель или управляемый объект управления назовем **универсальным заменителем** или **абсолютно управляемым** объектом, имеет ли место равенство

$$(\forall \langle s, s' \rangle)(\exists x)(\delta_o(s, x, s')),$$

в частности [3]

$$(\forall \langle s, s' \rangle)(\exists x)(\exists [z])(\delta_o(s, x, [z]) = s').$$

Пример 4. 1.: Найти принужденные траектории состояний в универсальном детерминированном заменителе (абсолютно управляемом детерминированном объекте), представленном параллельным регистром из двух элементов задержек, т.е. в таблице переходов заменителя (таб.4.1.с), если желаемый объект задан таблицей переходов (таб.4.1.а), причем кодирование состояний прототипа $\{1,2,3,4\} \rightarrow \{0,1\}^2 : 1 \mapsto 00, 2 \mapsto 01, 3 \mapsto 11, 4 \mapsto 10$, преобразует таб.4.1.а) в таб.4.1.б). \blacksquare

Таб. 4. 1.

a)

		s'	
$s \backslash x$	a	b	
→ 1	↓ 2	↓ 2	
2	↓ 3 ↓	↓ 3 ↓	
3	↓ 4	↓ 4 ↓	
4	↓ (4)	↓ 3 ↑	

b)

		$q'_1 q'_2$	
$q_1 q_2 \backslash x$	a	b	
0 0	0 1	0 1	
0 1	1 1	1 1	
1 1	1 0	1 0	
1 0	(1 0)	1 1	

c)

		$q'_1 q'_2$			
$q_1 q_2 \backslash D_1 D_2$	00	01	11	10	
→ 0 0	(0 0) $\xrightarrow{a, b}$	0 1	1 1	1 0	
0 1	0 0	$\downarrow a, b$ (0 1) $\xrightarrow{a, b}$	1 1	1 0	
1 1	0 0	0 1	$\downarrow a, b$ (1 1) $\xrightarrow{a, b}$	1 0	
1 0	0 0	0 1	$\uparrow b$ 1 1 \xleftarrow{b}	(1 0) $\downarrow a, b$	

Совпадают ли задуманные собственные траектории прототипа с собственными траекториями заменителя (при некотором кодировании состояний), то естественно $u = x$ и заменителем хватит управлять директивно.

5. Новая парадигма логического управления

Преобладающие технологические машины и устройства только кажутся динамическими объектами, ибо лица, идентифицирующие машины и устройства, неумышленно отождествляются с соответствующими концепционными управляющими автоматами, образуя, тем самым, совместно с технологическими устройствами или машинами, динамические системы управления.

О п р е д е л е н и е 5. 1. Объекты, в которых не выполняются собственные траектории состояний под воздействием лишь задающего управления, назовем **потенциально - динамические** [25,29].

Т е о р е м а 5. 1. В потенциально-динамических объектах выполняются собственные траектории тогда, когда потенциально-динамические объекты входят в состав системы управления (рис.3.1.b)) как объекты управления.

Д о к а з а т е л ь с т в о : Пусть $\hat{A} = \langle U \times Y \times [Z], X, \hat{\lambda}_A \rangle$, $N(\hat{A}, nO) = \langle U \times [Z], S, \delta_L, s_H \rangle$ суть конечно-автоматные модели соответственно искомого управляющего автомата (14) и системы управления (16) из рис.3.1.b) и пусть упорядоченная шестерка

$$nO = \langle X \times [Z], S, Y, \delta_{nO}, \Lambda_O, s_H \rangle \quad (19)$$

есть конечный, отнюдь не автомат, а псевдоавтомат, где δ_{nO} – отношение

$$\delta_{nO} = X \times S : \langle x, s' \rangle,$$

в частности функция

$$\delta_{nO} : X \times [Z] \rightarrow S : \langle x, [z] \rangle \mapsto s'$$

псевдопереходов, моделирующий потенциально-динамический объект управления. Действительно, по рис.3.1.b) имеет место равенство

$$\delta_L = \Lambda_O \circ \hat{\lambda}_A \circ \delta_{nO},$$

т.е.

$$(S \times U \times S) = (S \xrightarrow{\Lambda_O} Y) \circ (Y \times U \xrightarrow{\hat{\lambda}_A} X) \circ (X \times S),$$

в частности

$$(S \times U \times [Z] \xrightarrow{\delta_L} S) = (S \xrightarrow{\Lambda_O} Y) \circ (Y \times U \times [Z] \xrightarrow{\hat{\lambda}_A} X) \circ (X \times [Z] \xrightarrow{\delta_{nO}} S).$$

Поскольку для любого перехода $\delta_L(s, u, s')$, спец. $\delta_L(s, u, [z]) = s'$, каждой собственной траектории, выбранной задающим управлением u ($u \in U$), имеет силу

$$\text{proj}_3 \delta_L(s, u, s') = \text{proj}_2 \delta_{nO}(\lambda_A(\Lambda_O(s), u) s'),$$

спец.

$$\delta_L(s, u, [z]) = \delta_{nO}(\lambda_A(\Lambda_O(s), u, [z]), [z]),$$

то любой переход $\delta_L(s, u, s')$, спец. $\delta_L(s, u, [z]) = s'$, каждой собственной траектории, выбранной u , исполнен своим исходным состоянием s . Таким образом, выполнение любой собственной траектории в системе управления по рис.3.1.b) обеспечено. \square

Пример 5.1. Система воздуха [17] среднего давления состоит из клапана (К), компрессора (С) и манометра (М). Акторы клапана (k) и компрессора (c) являются одновходовыми и потенциальными (рис.5.1.). Состояние потенциально-динамического объекта управления – упорядоченная семерка $\tilde{K} \tilde{C} \tilde{N} \tilde{E} \tilde{T}_1 \tilde{T}_2 \tilde{T}_3$, где N – нормальная работа установки, E – неисправность работы, T_i – выдержка во времени ($i = 1, 2, 3$) с тем, что:

$$\tilde{C} = \begin{cases} \bar{C} - \text{выкл.} \\ C - \text{вкл.} \end{cases}, \quad \tilde{K} = \begin{cases} \bar{K} - \text{закрыт} \\ K - \text{открыт} \end{cases}, \quad \tilde{N} = \begin{cases} \bar{N} - \text{ненорм.} \\ N - \text{норм.} \end{cases}, \quad \tilde{E} = \begin{cases} \bar{E} - \text{исправность} \\ E - \text{неисправность} \end{cases},$$

$$\tilde{T}_i = \begin{cases} \bar{T}_i - \text{пока не истекло } T_i \\ T_i - \text{истекло } T_i \end{cases}. \quad \text{Машинное управление: } \tilde{k} = \begin{cases} \bar{k} - \text{закрыть } K \\ k - \text{открыть } K \end{cases},$$

$$\tilde{c} = \begin{cases} \bar{c} - \text{выключить } C \\ c - \text{включить } C \end{cases}, \tilde{n} = \begin{cases} \bar{n} - \text{сигнал } \bar{N} \\ n - \text{сигнал } N \end{cases}, \tilde{e} = \begin{cases} \bar{e} - \text{сигнал } \bar{E} \\ e - \text{сигнал } E \end{cases}, \tilde{t}_i = \begin{cases} \bar{t}_i - \text{пока не включает} \\ t_i - \text{уже включает} \end{cases}$$

Задающее управление: start, stop и R – разблокировка. При нажатии кнопки start открывается К. После его открытия включается С и через 5 с после пуска С закрывается К. Если через 20 с после пуска С давление М достигнет $M_{\text{норм}}$, т.е. если на УА воздействует

возмущение $\tilde{D} = \begin{cases} \bar{D} - M \leq M_{\text{норм}} \\ D - M \geq M_{\text{норм}} \end{cases}$, то горит лампочка N. В противном случае открывается

К, а затем горит лампочка E. Через 10 с после открытия К компрессор С останавливается и К закрывается. Последующий запуск системы управления возможен после снятия E нажатием кнопки R. При нажатии кнопки stop, когда С работает, открывается К. Через 10 с после открытия К останавливается С, а К закрывается. Диаграмма переходов системы управления имеется на рис. 5. 2. где $0 - \bar{K} \bar{C} (\bar{N} \bar{E}) \bar{T}_1 \bar{T}_2 \bar{T}_3$, $1 - K \bar{C} (\bar{N} \bar{E}) \bar{T}_1 \bar{T}_2 \bar{T}_3$, $2 - K C (\bar{N} \bar{E}) \bar{T}_1 \bar{T}_2 T_3$, $3 - \bar{K} C (\bar{N} \bar{E}) \bar{T}_1 \bar{T}_2 T_3$, $4 - \bar{K} C (N \bar{E}) \bar{T}_1 \bar{T}_2 \bar{T}_3$, $5 - K C (\bar{N} E) \bar{T}_1 T_2 \bar{T}_3$, $6 - K C (N \bar{E}) \bar{T}_1 T_2 \bar{T}_3$, $7 - \bar{K} \bar{C} (\bar{N} E) \bar{T}_1 \bar{T}_2 \bar{T}_3$, $8 - \bar{K} \bar{C} (N E) \bar{T}_1 \bar{T}_2 \bar{T}_3$. Таб. 5. 1. – таблица выходов управляющего автомата. Отметим, что срабатывание некоторых частей системы воздуха среднего давления через определенное время осуществляется посредством динамических функциональных элементов задержки (таймеров) – см. приложение 3. Таймерами управляют лишь директивно, а обратные связи, в которых таймеры находятся, не служат для управления ими, а лишь для подачи осведомляющих сигналов об истечении заданного времени в управляющий автомат. ■

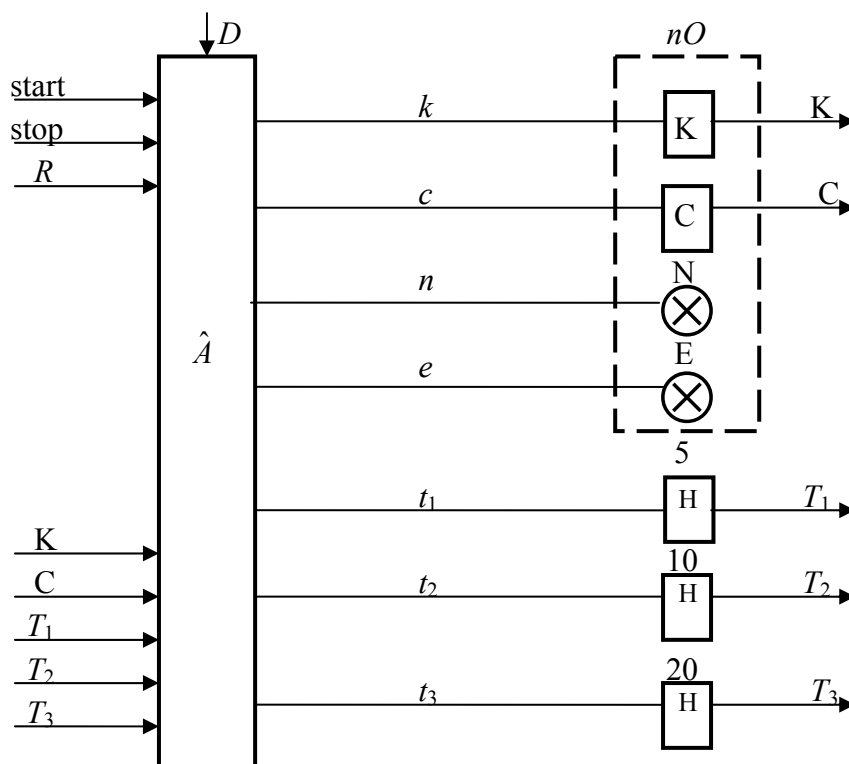


Рис. 5. 1. Система воздуха среднего давления из примера 5.1.

Таб. 5.1.

u, z	start			D	\bar{D}	stop			R	
s	0	1	2	3	3	4	5	6	7	8
x	k	c	$t_1 t_3$	$t_3 n$	$t_3 e$	kn	$\bar{k} \bar{c} t_2 e$	$\bar{k} \bar{c} t_2 n$	\bar{e}	\bar{n}

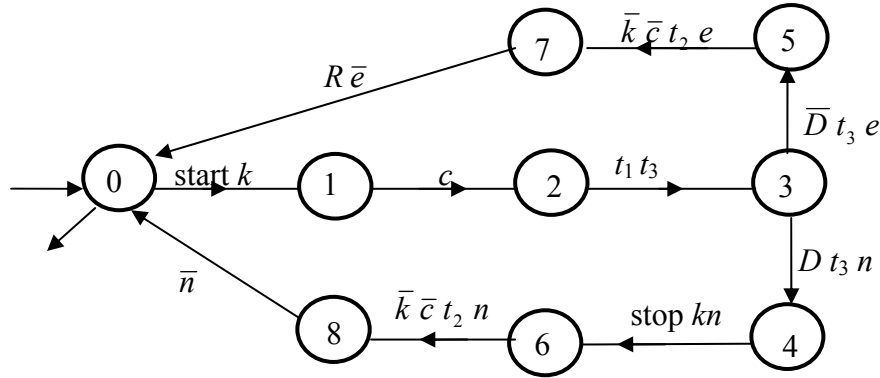


Рис. 5. 2. Диаграмма переходов системы управления из примера 5.1.

Отношение δ_{nO} псевдопереходов в nO можно, как неинтересное, игнорировать, а функция λ_A выходов (машинного управления) получается просто из конечно-автоматной модели системы управления.

Требуется ли, чтобы эволюция системы $N(\hat{A}, nO)$ (16) протекала вдоль принужденных траекторий состояний, строят систему $N(\hat{A}_h, N(\hat{A}, nO))$ – рис. 3. 1. с), где \hat{A}_h – иерархически выше стоящий управляющий автоиат, управляет который динамической системой $N(\hat{A}, nO)$.

Пример 5. 2: Пусть дана квадратная игральная коробка длиной в d и три игральные кости длиной в $d/2$ (рис.5.2.a)). Стараемся освободить левую верхнюю клетку коробки, захвачивая, пока пустую, нижнюю клетку, передвигая кости вниз или направо через отдельные клетки 1,2,3,4 вдоль траекторий или $\alpha : 4 \rightarrow 2 \rightarrow 1$, или $\beta : 4 \rightarrow 3 \rightarrow 1$. Свободную клетку примем за состояние конечно-автоматной модели коробки. Не полагается ли любезный читатель на сомнительный телекинез, ясно, что передвижение костей в коробке обеспечивает собственноручно статический игрок1 (рис.5.2.c)). Моделью динамической системы $N(\text{игрок1}, \text{коробка})$ является полуавтомат с функцией переходов по таб.5.2.a) с помеченными собственными траекториями состояний. Отсюда таблица выходов игрока1 (таб.5.2.b)) и таблица псевдопереходов коробки (таб.5.2.c)). Функция переходов системы $N(\text{игрок1}, \text{коробка})$ с помеченными принужденными траекториями состояний α, β (α - сплошная, β - штрихованная линия) имеется в таб.5.2.d). Пусть игрок1 и над ним вышестоящий игрок2 образуют двухединного игрока (рис.5.2.b)). Модель системы $N(\text{игрок2}, N(\text{игрок1}, \text{коробка}))$, есть функция переходов (таб.5.2.e)), где stop, start α , start β – управление задаваемое игроку2 распорядителем. Функция выходов игрока2 имеется в таб. 5.2.f). ■

Таб. 5. 2.

a)

$s \backslash v$		s'	
		вниз	направо
1	↑	(1)	↑ (1)
2	↑	(2)	↑ 1
3	↑	1	(3) ↑
4	↑	2	3 ↑

b)

v	вниз	вниз	направо	направо
s	3	4	2	4
x	a	b	c	d

c)

x	a	b	c	d
s	3	4	2	4

d)

$s \backslash v$		s'	
		вниз	направо
1	↑	(1)	(1) ↑
2	↑	(2)	↑ 1
3	↑	1	(3) ↑
4	↑	2	3 ↑

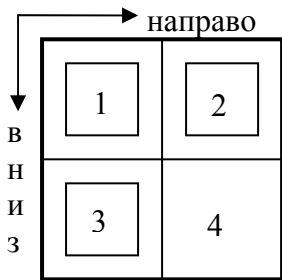
e)

$s \backslash u$		s'		
		stop	start α	start β
1	(1)	(1)	(1)	
2	(2)	(2)	1	
3	(3)	1	(3)	
4	(4)	2	3	

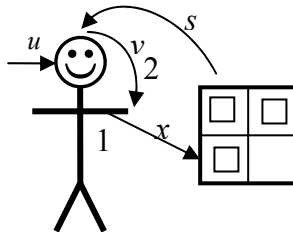
f)

$s \backslash u$		s'			v		
		stop	start α	start β	stop	start α	start β
1	(1)	(1)	(1)	—	—	—	
2	(2)	(2)	1	—	направо	—	
3	(3)	1	(3)	—	—	вниз	
4	(4)	2	3	—	вниз	направо	

a)



b)



c)

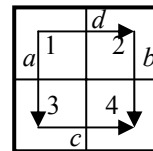


Рис. 5. 2. а) Игральная коробка, б) система N (игрок1.коробка), в) работа игрока 1 из примера 5. 2.

6. Программное логическое управление

Не имея возможность определить текущее состояние детерминированного динамического или потенциально-динамического объекта управления, прибегают к т.н. **программному** управлению по разомкнутому контуру (рис. 6.1.)

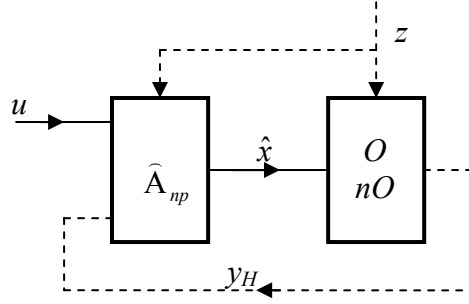


Рис. 6.1. Программное управление.

Программное управление допускает [25] воздействие на систему управления явных возмущений и считает, что получение реакции y или ее своевременная обработка либо невозможна, либо слишком дорого обходится (за исключением y_H).

Рассмотрим любой переход между состояниями собственной или принужденной траектории состояний v , с одной стороны, динамическом заместителе задуманного динамического объекта управления $\delta_O(s, x, [z]) = s'(k(\delta_T(q, u, [z]))) = \delta_L(k(q), u, [z]) = \delta_O(s, x, [z])$, и с другой, в динамической системе управления потенциально-динамическим объектом $\delta_L(s, u, [z]) = s'(\delta_L(s, u, [z]) = \delta_{nO}(x, [z]))$. Свяжем с моментами t_i, t_{i+1} (см. раздел 2) пребывание заместителя или системы управления в состоянии соответственно s, s' при воздействии машинного управления x и при заданном задающем управлении u .

О п р е д е л е н и е 6. 1. Назовем разницу $\Delta t = t_{i+1} - t_i$ при $\tau : S^2 \rightarrow 2^{\mathbf{R}_0^+} : \langle s, s' \rangle \mapsto [t_i, t_{i+1})$ и $\tau : X \rightarrow 2^{\mathbf{R}_0^+} : x \mapsto [t_i, t_{i+1})$, где \mathbf{R}_0^+ – множество неотрицательных действительных чисел, **продолжительностью** соответственно перехода из s в s' и машинного управления x с тем, что $\hat{S} : S^2 \rightarrow (\mathbf{R}_0^+)^2 : (\langle s, s' \rangle \mapsto \langle t_i, t_{i+1} \rangle) = (s, s')_{t_i}^{t_{i+1}}$ и $\hat{X} : X \rightarrow (\mathbf{R}_0^+) : (x \mapsto \langle t_i, t_{i+1} \rangle) = x_{t_i}^{t_{i+1}}$.

Предположим, что $\tau(s, s') = \tau(x)$ и что располагаем оценками продолжительностей всех переходов любой траектории.

Т е о р е м а 6. 1. Моделью управляющего автомата \hat{A}_{np} служит упорядоченная тройка

$$\hat{A}_{np} = \langle U \times [Z], \hat{S}, \hat{X}, \lambda_{\hat{A}_{np}} \rangle,$$

где $\lambda_{\hat{A}_{np}}$ – функция выходов

$$\lambda_{\hat{A}_{np}} : U \times [Z] \times \hat{S} \rightarrow \hat{X} : \langle u, [z], (s, s')_{t_i}^{t_{i+1}} \rangle \mapsto x_{t_i}^{t_{i+1}}$$

при $\Delta t_i = t_{i+1} - t_i$.

Д о к а з а т е л ь с т в о : Пусть задана, без ущерба для общности, собственная траектория состояний

$$T_i = (\{s_H\} \times \{u\} \times [Z] \rightarrow S) \circ (S \times \{u\} \times [Z] \rightarrow S) \circ \dots \circ (S \times \{u\} \times [Z] \rightarrow \{s_{Ki}\})$$

и пусть заданы $\tau(s_H, s_0) = \tau(\Lambda_O^{-1}(y_H), \Lambda_O^{-1}(y_0)) = \Delta t_H$, $\tau(s_0, s_1) = \tau(\Lambda_O^{-1}(y_0), \Lambda_O^{-1}(y_1)) = \Delta t_0, \dots$, $\tau(s_{K_{i-1}}, s_{K_i}) = \tau(\Lambda_O^{-1}(y_{K_{i-1}}), \Lambda_O^{-1}(y_{K_i})) = \Delta t_{K_{i-1}}$, где, с одной стороны, $\delta_O(s_H, x_0, [z_0]) = s_0 = k(\delta_T(q_H, u, [z_0]))$, $\delta_O(s_0, x_1, [z_1]) = s_1 = k(\delta_T(q_0, u, [z_1]))$, ..., $\delta_O(s_{i,f-1}, x_{if}, [z_{if}]) = s_{K_i} = k(\delta_T(q_{i,f-1}, u, [z_{if}]))$, и с другой, $\delta_L(s_H, u, [z_0]) = s_0 = \delta_{nO}(x_0, [z_0])$, $\delta_L(s_0, u, [z_1]) = s_1 = \delta_{nO}(x_1, [z_1])$, ..., $\delta_L(s_{i,f-1}, u, [z_{if}]) = s_{K_i} = \delta_{nO}(x_{if}, [z_{if}])$. Отсюда $\lambda_{\lambda_{A_{np}}}(u, [z_0], (s_H, s_0)_{t_H}^{t_0}) = x_{t_H}^{z_0}$, $\lambda_{\lambda_{A_{np}}}(u, [z_1], (s_0, s_1)_{t_0}^{t_1}) = x_{t_0}^{z_1}, \dots, \lambda_{\lambda_{A_{np}}}(u, [z_{i,f}], (s_{i,f-1}, s_{K_i})_{t_{i,f-1}}^{t_{if}}) = x_{t_{i,f-1}}^{z_{if}}$. ■

Итак, программный управляющий автомат, управляет который либо двоичным, с унарно кодированными состояниями, детерминированным $O(15)$, либо $nO(19)$ в детерминированной системе управления, представляет собой цепочку (рис.6.2.) таймеров-одноходовых (ждущих) мультивибраторов таких, что $kRC = \Delta t_i$, где k – постоянная. Как правило, логическое управление встречается в смешанном виде как по разомкнутому, так и по замкнутому контурам. Особенно важен т.н. **временной контроль** выразительного достоверного перехода между состояниями, сравнивает который желаемое достижение состояния s' с истекшим временем Δt_i ; двоичная схема временного контроля приводится на рис.6.2. б).

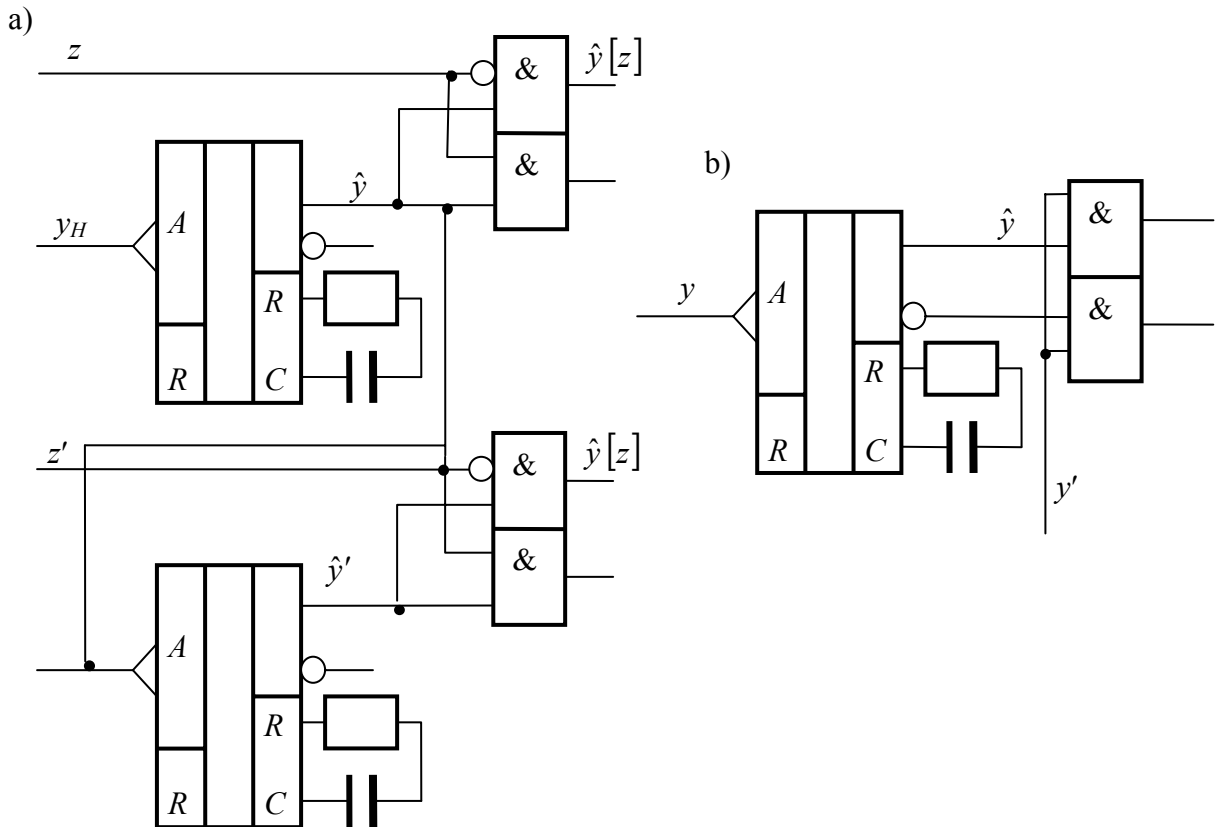


Рис. 6 2. а) Фрагмент цепочки таймеров, выполняющих переход $s' = \delta_O(s, x, z)$ или $s' = \delta_L(s, x, z)$ в двоичном динамическом объекте или потенциально-динамическом объекте, б) двоичный модуль временного контроля.

7. Оптимальное логическое управление

Рассматривая собственные траектории состояний в концепционном динамическом объекте, заменителем которого собираемся управлять, или в системе управления потенциально-динамическим объектом, будем вместо отношения, в частности функции, переходов соответственно $\delta_T(q, u, q')$, спец. $\delta_T(q, u, [z]) = q'$, или $\delta_L(s, u, s')$, спец. $\delta_L(s, u, [z]) = s'$ писать, ради краткости, $\delta(s, u, s')$, спец. $\delta(s, u, [z]) = s'$.

О п р е д е л е н и е 2. 2. Под **системой** либо **возможных**, либо **факультативных переходов** в (2)/(3) при заданных состояниях s и машинном управлении x будем соответственно подразумевать либо:

$$\delta : S \times U \times S : \langle s, u, s' \rangle, \text{ спец. } \delta : S \times U \times [Z] \rightarrow S : \langle s, u, [z] \rangle \mapsto s',$$

либо:

$$\delta : S \times U \times \{ЛПР\} \rightarrow S : \langle s, u, ЛПР \rangle \mapsto s',$$

где *ЛПР* – целеустремленное альтернативное решение лица, принимающего решения. Отдельные переходы системы или возможных, или факультативных переходов назовем соответственно переходами либо **случайными**, либо **конкурентными**.

В (2)/(3) могут встречаться как случайные, так конкурентные переходы между состояниями. Предполагается взаимная независимость всех случайных переходов в любой системе возможных переходов в (2)/(3).

О п р е д е л е н и е 7. 1. Траектории состояний с начальным состоянием s_H и конечными состояниями s_{K_j} , содержат которые хотя бы один конкурентный переход между состояниями, назовем **конкурентными**.

Говоря об отрезке конкурентной траектории, имеем в виду ее подтраекторию.

Рассмотрим вероятностную модель системы возможных переходов, спец. достоверный переход, случайные переходы которой суть $s' \in \text{proj}_3 \{ \delta(s, u, s') \}$, в частности $s' = \delta(s, u, [z])$. Тогда случайный переход вероятностно моделируется (условной) вероятностью перехода $p(s' | s, u)$ и вся система возможных переходов моделируется распределением $P_s = \{ p(s' | s, u) \}_{s'}$ ($\sum_{s'} p(s' | s, u) = 1$). Учитывая ожидаемое воздействие случайных измеримых возмущений $z \in Z$, или-же при заданном разбиении $\mathcal{Z} = \{ \mathcal{z} \}_{\mathcal{z} \in \mathcal{Z}}$ на Z , т.е. $z \in \mathcal{z}$, то $p(s' | s, u) = p(\mathcal{z})$, является ли z достаточной причиной (вместе с задающим управлением u) вызова перехода из s в одно из возможных s' . Для достоверного перехода имеет силу $p(s' | s, u, [z]) = 1$.

Предпочитаем вероятностную меру, т.е. численную оценку ожидания (удовлетворительного предсказания), осуществления перехода, или воздействия измеримого возмущения, ибо она способна моделировать сопряженность состояний в траекториях состояний – равенство Колмогорова и Чепмена (Chapman).

Пусть T_r конечное множество всех состоянческих переходов, обозначенных, ради простоты, через $t_r (t_r \in T_r)$, собственной траектории состояний (2)/(3).

Определение 7.2. Гомоморфизм $u: T_r \rightarrow \mathbf{R}_0^+ : t_r \mapsto u$ или, не теряя общность, $u: T_r \circ T_r \rightarrow \mathbf{R}_0^+ : \langle t_{ri}, t_{rj} \rangle \mapsto u$, где t_{ri}, t_{rj} – сопряженные состоянческие переходы)*, из $\langle T_r, \prec, \circ \rangle$ в $\langle \mathbf{R}_0^+, <, + \rangle$, где $\prec: T_r^2 : t_{ri} \prec t_{rj}$ наблюдаемое отношение „ t_{rj} предпочтительнее, чем t_{ri} “, т.е.

$$t_{ri} \prec t_{rj} \Rightarrow u(t_{ri}) < u(t_{rj}); u(t_{ri}, t_{rj}) = u(t_{ri}) + u(t_{rj}),$$

называется аддитивной **функцией полезности** и $u = u(t_r)$ – **полезность** перехода t_r .

Теорема 7.1. Функция полезности удовлетворяет условию

$$t_{ri} \prec t_{rj} \Leftrightarrow u(t_{ri}) < u(t_{rj})$$

тогда и только тогда, когда отношение \prec асимметрично и отрицательно транзитивно.)**

Доказательство: см. [30,31]. □

Следствие 7.1. Отношение \prec транзитивно.

Пусть имеется диаграмма переходов системы факультативных или возможных переходов, вершины и ребра которой помечены полезностями соответственно исходного состояния $u(s)$ и переходов u_i ($i = 1, 2, \dots, k$) – рис.7.1.

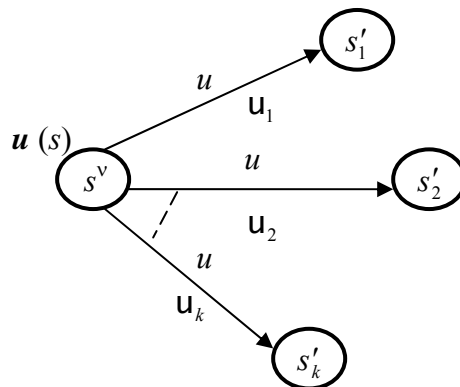


Рис. 7.1. Система состоянческих переходов.

Теорема 7.2. Для полезности $u(s'_i)$ ($i = 1, 2, \dots, k$) состояния-преемника s'_i из рис.7.1. имеет место равенство, двигаясь в диаграмме слева направо

$$u(s'_i) = u(s) + u_i$$

)* Если $t_{ri} = \delta(s, u, s')$ и $t_{rj} = \delta(s', u', s'')$, то $\langle t_{ri}, t_{rj} \rangle = \delta(s, u u', s'')$

)** - $t_i \prec t_j \Rightarrow \text{non}(t_j \prec t_i)$ – асимметричное,

- $\text{non}(t_i \prec t_j) \wedge \text{non}(t_j \prec t_k) \Rightarrow \text{non}(t_i \prec t_k)$ ($t_i, t_j, t_k \in T$) – отрицательно транзитивное отношение.

Доказательство: Ведь исходное состояние s в системе переходов из рис.7.1. есть необходимая, а также исполнительная причина вызова любого перехода из рис.7.1. [25,29]. Таким образом, полезность $u(s'_i)$ является итоговой полезностью созданной полезностью $u(s)$ исходного состояния s и полезностью u_i перехода. \square

Тем самым, можно определить итоговую полезность или итоговые затраты (качество) каждой конкурентной траектории состояний из (2)/(3). Двигаясь в сетевой диаграмме слева направо или справа налево по принципу оптимальности (см. ниже), не встречаются ли в собственных траекториях состояний системы возможных переходов или двигаясь справа налево по принципу оптимальности Беллмана (см. ниже) в обратном случае, вполне оправданы следующие определения.

Определение 7.3. Под ожидаемым (средним) значением полезности исходного состояния s системы возможных переходов $\delta(s, u, s')$ подразумевают

$$u(s) = \sum_i (u(s'_i) + u_i) p(s'_i | s, u).$$

Определение 7.4. Под оценкой полезности исходного состояния s системы факультативных переходов $\delta(s, u, \text{ЛПР}) = s'$ можно понимать смесь Гурвица (Hurwitz)

$$u(s) = \alpha \max_i \{u(s'_i) + u_i\} + (1 - \alpha) \min_i \{u(s'_i) + u_i\},$$

где α ($\alpha \in [0,1]$) оценивает отношение субъекта к конкурентным переходам с безграничного оптимизма ($\alpha = 1$) по крайний пессимизм ($\alpha = 0$).)*

По определениям 7.3., 7.4. можно постановить полезность или затраты любого состоянческого перехода в (2)/(3). Итоговая полезность – **качество конкурентной траектории состояний** – представляет собой, без излишнего формализма, сумму полезностей или ожидаемых полезностей всех систем соответственно конкурентных или случайных переходов, содержащихся в траектории.

Определение 7.5. Конкурентная траектория состояний в заданной собственной траектории (2)/(3) называется **оптимальной**, если ее качество по некоторому критерию экстремально на множестве всех конкурентных траекторий в (2)/(3). Естественно назвать управляющий автомат, обеспечивающий в объекте управления или в системе управления выполнение оптимальной траектории состояний, **оптимальным** автоматом.

В принципе, можно отыскать оптимальную конкурентную траекторию состояний простым перебором всех конкурентных траекторий в (2)/(3), что становится непрактичным для сложных (2)/(3). Следующий **принцип оптимальности** доставляет более простую процедуру, чем простой перебор.

Теорема 7.3. Любой отрезок оптимальной конкурентной траектории состояний в (2)/(3) оптимален [32].

)* Оценку α или $(1 - \alpha)$ можно трактовать как вероятность того, что субъект склонен учитывать соответственно скорее $\max \{ \sum u(\delta) \}$ или $\min \{ \sum u(\delta) \}$. Могут встречаться более подходящие формальные модели поведения субъекта; автор, конечно, не психолог и поэтому ограничился приведенной смесью.

Доказательство: Рассмотрим любой отрезок оптимальной траектории. Можно ли его заменить в данной траектории отрезком, качество которого выше или ниже чем рассматриваемого отрезка, то качество возникшей траектории станет выше или ниже чем траектории оптимальной, что противоречит предположению теоремы. \square

Когда в (2)/(3) встречаются системы факультативных переходов, заметим, что в каждой системе факультативных переходов (2)/(3) исследуются при нахождении оптимальной траектории сначала все конкурентные переходы, а затем выбирается в любой системе факультативных переходов один надлежащий. Во избежание возможных вычислительных трудностей пользуются вместо принципом оптимальности его частным Беллмановым случаем: любой конечный отрезок оптимальной конкурентной траектории состояний оптимален. Заметим, что существуют ли в собственной траектории лишь системы возможных переходов, то траектория является оптимальной.

Как следствие теоремы 7.2. приводится утверждение [32] о том, что оптимальная траектория не содержит циклы, ибо в обратном случае [33], переходы цикла составляли бы порочный круг и никогда не могли бы начаться.

Определение 7.6. Под **сетевой диаграммой траекторий** (2)/(3) понимают диаграмму переходов в (2)/(3), ребра которой помечены полезностью или затратами состоянческих переходов.

Пример 7.1. Пусть задана сетевая диаграмма (рис.7.2.а)), ребра которой помечены полезностями [33]. Найти оптимальную траекторию в заданной сетевой диаграмме и оптимальный управляющий автомат с тем, что субъект абсолютный оптимист ($\alpha = 1$). Воспользовавшись принципом оптимальности, оцениваем самой большой полезностью $u(i)$ состояния i ($i = 1, 2, \dots, 8$), начиная с либо конечного 8, либо с начального (1) состояния, т.е.

$$\begin{aligned}
 u(8) &= /0/; & u(1) &= (0); \\
 u(5) &= u(8) + 5 = /5/; & u(2) &= u(1) + 5 = (5); \\
 u(6) &= \max \{u(5) + 4, u(8) + 7\} = /9/; & u(3) &= u(1) + 7 = (7); \\
 u(7) &= u(8) + 9 = /9/; & u(4) &= \max \{u(1) + 8, u(3) + 3\} = (10); \\
 u(2) &= u(5) + 1 = /6/; & u(5) &= \max \{u(2) + 1, u(3) + 9, u(6) + 4\} = (20); \\
 u(4) &= \max \{u(6) + 6, u(7) + 4\} = /15/; & u(6) &= \max \{u(3) + 3, u(4) + 6\} = (16); \\
 u(3) &= \max \{u(4) + 3, u(5) + 9, u(6) + 6\} = /18/; & u(7) &= u(4) + 4 = (14); \\
 u(1) &= \max \{u(2) + 5, u(3) + 7, u(4) + 8\} = /25/. & u(8) &= \max \{u(5) + 5, u(6) + 7, u(7) + 9\} = (25).
 \end{aligned}$$

Оптимальная траектория обозначена на рис.7.2.а) жирной линией и функция выходов оптимального управляющего автомата имеется в таб.7.1.а). Предположим, ради простоты, что „настроение“ субъекта оценено $\alpha = 0,5$. Тогда по Беллману:

$$\begin{aligned}
 u(8) &= 0; \\
 u(7) &= u(8) + 9 = 9; \\
 u(5) &= u(8) + 5 = 5; \\
 u(6) &= 0,5 \max \{u(5) + 4, u(8) + 7\} + 0,5 \min \{u(5) + 4, u(8) + 7\} = 8; \\
 u(2) &= u(5) + 1 = 6; \\
 u(4) &= 0,5 \max \{u(6) + 6, u(7) + 4\} + 0,5 \min \{u(6) + 6, u(7) + 4\} = 13,5; \\
 u(3) &= 0,5 \max \{u(4) + 3, u(5) + 9, u(6) + 6\} + 0,5 \min \{u(4) + 3, u(5) + 9, u(6) + 6\} = 13,75 \\
 u(1) &= 0,5 \max \{u(2) + 5, u(3) + 7, u(4) + 8\} + 0,5 \min \{u(2) + 5, u(3) + 7, u(4) + 8\} = 16,25
 \end{aligned}$$

Оптимальная траектория на рис.7.2.б) обозначена жирной линией и функция выходов оптимального управляющего автомата имеется в таб. 7.2.б). ■

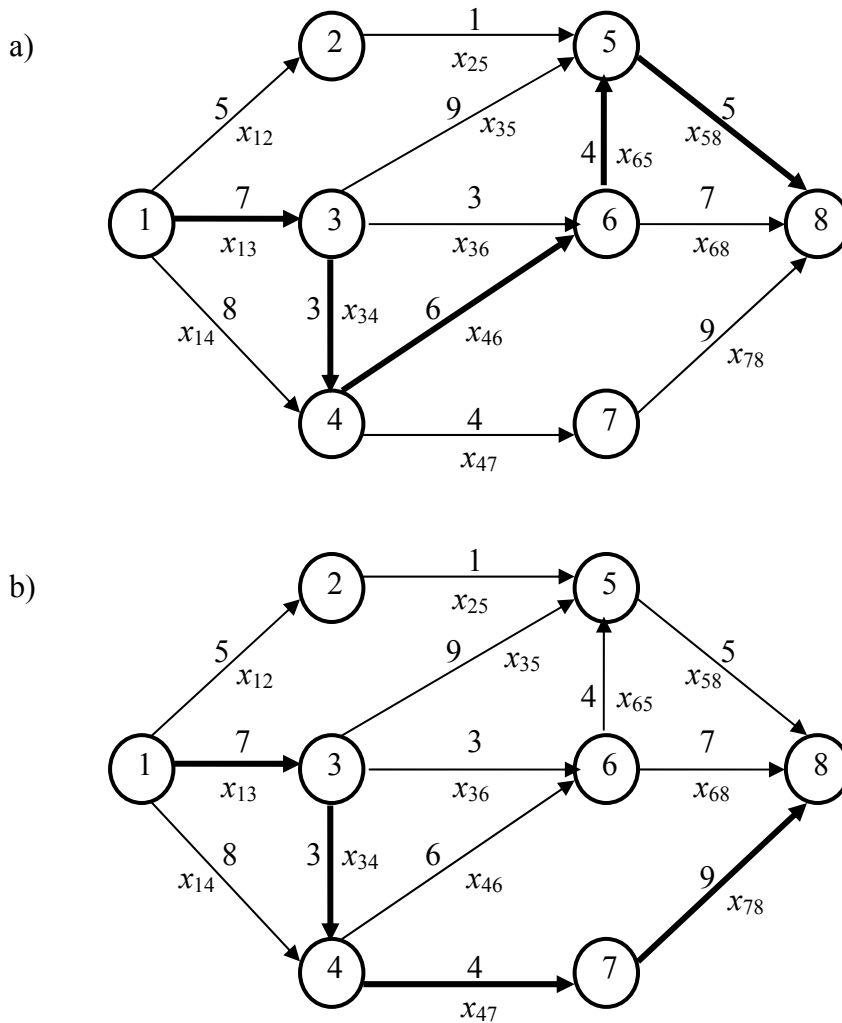


Рис. 7. 2. а), б) Оптимальные траектории из примера 7.1.

Таб. 7. 1.

a)

s	1	3	4	5	6
x	x_{13}	x_{34}	x_{46}	x_{58}	x_{65}

b)

s	1	3	4	7
x	x_{13}	x_{34}	x_{47}	x_{78}

Пример 7. 2. Построить оптимальную траекторию в сетевой диаграмме из рис.7.3. и оптимальный управляющий автомат, причем системы переходов в состояниях 3 и 6 суть возможные с распределением вероятностей $P_3 = \{p(5|3, u_3) = 0,2; p(6|3, u_3) = 0,3; p(4|3, u_3) = 0,5\}$ и $P_6 = \{p(5|6, u_6) = 0,4; p(8|6, u_6) = 0,6\}$. Отсюда

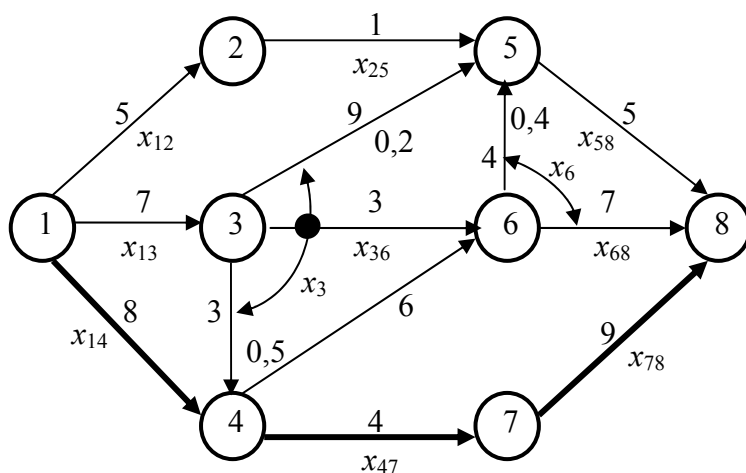


Рис. 7. 3. Оптимальная траектория из примера 7. 2.

$$\begin{aligned}
 u(8) &= 0; \\
 u(5) &= u(8) + 5 = 5; \\
 u(6) &= \max \{ (u(5) + 4) 0,4; (u(8) + 7) 0,6 \} = 4,2; \\
 u(7) &= u(8) + 9 = 9; \\
 u(2) &= u(5) + 1 = 6; \\
 u(4) &= \max \{ u(6) + 6, u(7) + 4 \} = 13; \\
 u(3) &= \max \{ (u(4) + 3) 0,5; (u(5) + 9) 0,2; (u(6) + 3) 0,3 \} = 9; \\
 u(1) &= \max \{ u(2) + 5, u(3) + 7, u(4) + 8 \} = 21.
 \end{aligned}$$

Оптимальная траектория обозначена на рис.7.3. жирной линией и функция выходов оптимального управляющего автомата находится в таб.7.2. ■

Таб. 7. 2.

s	1	4	7
x	x_{14}	x_{47}	x_{78}

Само собой разумеется, что оптимальная траектория в (2)/(3) существует в общности не только одна.

Различают три разных подхода к многокритериальной оптимизации (2)/(3). Учитывая разные критерии k_i ($i = 1, 2, \dots, l$), по которым определяется качество конкурентных траекторий состояний, можно:

- построить по Парето (Pareto) для всех недоминирующих признаков k_i оптимальные траектории, ибо увеличение качества траектории состояний по одному из критериев сопровождается уменьшением (строго говоря неувеличением) качества траектории по другим критериям;

- построить композицию $\chi : \times_{i=1}^l \{u_i\} \rightarrow R_0^+ : \langle u_1, u_2, \dots, u_l \rangle \mapsto u$, где u_i – качество траектории по признаку i ; например взвешенную аддитивную $\sum_{i=1}^l u_i w_i$ или мультипликативную $\prod_{i=1}^l u_i^{w_i}$ свертку, где $w_i \in [0,1]$ и $\sum_{i=1}^l w_i = 1$, сохраняющую нуль, т.е. $\chi(0, 0,$

..., 0) = 0 и монотонную, т.е. $\langle \mathbf{u}_{i1}, \mathbf{u}_{i2}, \dots, \mathbf{u}_{il} \rangle \leq \langle \mathbf{u}_{j1}, \mathbf{u}_{j2}, \dots, \mathbf{u}_{jl} \rangle \Rightarrow \chi \langle \mathbf{u}_{i1}, \mathbf{u}_{i2}, \dots, \mathbf{u}_{il} \rangle \leq \chi \langle \mathbf{u}_{j1}, \mathbf{u}_{j2}, \dots, \mathbf{u}_{jl} \rangle$;

- упорядочить критерии k_i по отношению предпочтения и найти оптимальную траекторию по наиболее предпочтительному критерию; имеет ли выбор оптимальной траектории более чем одно решение „воспользоваться“ следующим по предпочтению признаком до того, пока мы или не используем все критерии, или не получим после продолжающегося применения признаков единственную оптимальную траекторию.

8. Адаптивное логическое управление

Рассмотрим (2)/(3), в которой встречаются вместе с конкурентными также системы возможных переходов, но с неизвестными распределениями вероятностей случайных переходов. Можно предложить четыре подхода к задаче разыскания оптимальных траекторий в (2)/(3):

- считать все случайные переходы конкурентными,
- считать все конкурентные переходы случайными и распределения вероятностей во всех системах возможных переходов принимать за равномерные;
- определить после неограниченного (практически конечного) количества повторяющихся выполнений (2)/(3) распределения статистических вероятностей переходов во всех системах возможных переходов;
- обратиться к параметрической – альтернативной адаптации, считая все встречающиеся системы конкурентных переходов возможными.

Займемся далее адаптивным логическим управлением, осознавая не только, что распределения вероятностей случайных переходов в системах возможных переходов не известны, но, что нет возможности их экспериментально определить. Стараемся поэтому воспользоваться вероятностями полученными в течение управления .

В адаптивной системе управления [34] актуальное состояние объекта управления s сравнивается с оптимальным состоянием s_{opt} и наблюдаемый результат их сравнения используется для определения способа, как достичь требуемого состояния (см. ниже). Адаптивную систему управления можно расширить хранением результатов экспериментов при управлении в адаптивной системе управления – обучающаяся система управления. В статье займемся случаем, когда обучающийся стохастический конечный автомат с переменной структурой (автоматы с жесткой структурой распалогают во много большим количеством состояний [35], чем автоматы со структурой переменной) моделирует ученика, входившего в состав объекта управления.

Ведь автоматный подход к обучению становится понятным на примере ученика и учителя [35]. Задавая ученику вопрос, при заданном конечном множестве альтернативных ответов, ученик может выбрать один из них. Учитель оценивает ответ как правильный или как плохой с тем, что оценивает качество ответов случайно, а с ненулевой вероятностью может решить, является ли ответ ученика правильным или плохим. Выход из, кажется безнадежного положения, состоит в том, что вероятность плохой оценки ответа самая низкая, если ответ правильный. В данной ситуации интересно найти способ, по которому может ученик получить сведения учителя такие, чтобы научиться правильно отвечать. Стараемся дойти до целесообразных или оптимальных ответов.

Рассмотрим бегло композицию адаптивной системы управления с подкрепляющим обучением, свободно следуя за [36] (рис.8.1.), где УА – управляющий автомат и ОУ – объект управления, ОА – обучающийся автомат, ИК – инспектор по качеству, оценивающий состояние объекта управления и НУ – несовершенный учитель, награждающий приращение оценок состояний объекта управления поощрением обучающегося автомата, указывая, является ли состояние s оптимальным. Предполагается, что конкурентные переходы между состояниями суть также случайные, или-же псевдослучайные и что (2)/(3) содержит одно конечное состояние s_K .

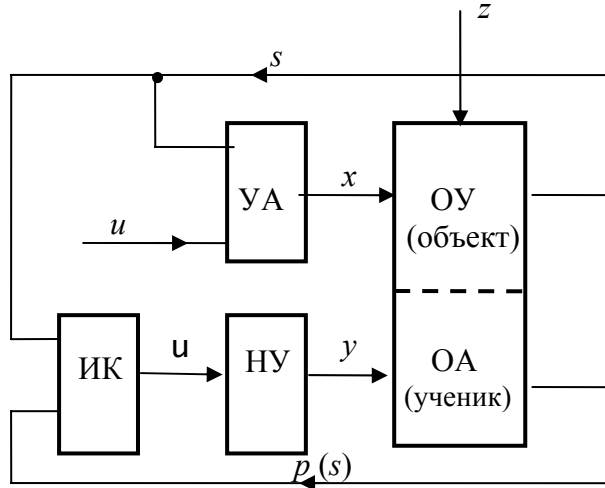


Рис. 8. 1. Блок-схема адаптивной системы логического управления с обучением.

Конечно-автоматные модели управляющего автомата и объекта управления уже приведены выше.

Моделью обучающегося автомата (ученика) является стохастический конечный автомат

$$OA = \langle Y, S, T, \Pi_H \rangle \quad (20)$$

где Y ($Y = \{0,1\}$), S – алфавит соответственно входной (поощрений), состояний, T – индикаторная функция

$$T : ([0,1])^{|S|} \times S \times Y \rightarrow ([0,1])^{|S|} : \langle \Pi, s, y \rangle \rightarrow \Pi'$$

с тем, что Π – распределение вероятностей возможного существования объекта в состояниях s_i при $i = 1, 2, \dots, |S|$, $\Pi : \{1, 2, \dots, |S|\} \rightarrow [0,1] : \{p(s_1), p(s_2), \dots, p(s_{|S|})\}$, причем $p : S \rightarrow [0,1] : s \mapsto p(s)$, $\sum_{s \in S} p(s) = 1$ и Π_H – начальное равномерное распределение

$\left\{ p(s_1) = \frac{1}{|S|}, p(s_2) = \frac{1}{|S|}, \dots, p(s_{|S|}) = \frac{1}{|S|} \right\}$. Отклик u учителя характеризует эффективность

функционирования обучающегося автомата, причем нештраф (0) или штраф (1) соответствует позитивной или негативной реакции объекта на машинное управление x . Пусть имеется распределение Π , т.е. наблюдателем (заказчиком) ожидаемые оценки того, что объект управления и, тем самым, обучающийся автомат будет находиться в одном из возможных состояний s и пусть актуальное состояние OA (20) и, тем самым, объекта управления есть s . OA (20) под воздействием мотивировки u попадает в одно из возможных состояний-преемников s' , вырабатывая распределение-преемников Π' . Воспринимая y ,

действие объекта продолжается. Значит, подкрепляющее обучение OA (20) заключается в том, что вероятность $p(s')$ при индикаторной функции вида: с одной стороны, „награждение-штраф“ в случае нештрафа ($y = 0$) увеличивается, а наоборот, уменьшается в случае штрафа ($y = 1$), и с другой, „награждение-бездеятельность“ $p(s')$ увеличивается при $y = 0$, а остается неизменной при $y = 1$. Исчерпывающую классификацию индикаторных функций можно найти в [35].

Модель объекта управления с подкрепляющим обучением имеет, таким образом, двойственный (дуальный) характер: с одной стороны, машинное управление x изменяет состояние объекта для достижения оптимальной траектории состояний в объекте, а с другой, несовершенный учитель возмущает поощрениями y ученика, чтобы получить вероятность $p(s)$ состояния s , затрагивая структуру ученика.

Моделью индикатора качества является упорядоченная тройка

$$I = \langle S \times [0, 1], \mathbf{R}_0^+, \mathbf{u} \rangle, \quad (21)$$

где функция полезности имеет форму $\mathbf{u} : S \times [0, 1] \rightarrow (\mathbf{R}_0^+)$: $\langle s, p(s) \rangle \mapsto \mathbf{u}$.

Обозначим через s_{opt} состояния оптимальной траектории в (2)/(3); тогда

$$s_{opt} = \arg \mathbf{u}(s_{opt}) = \arg \max_{s \in S} \{\mathbf{u}(s)\}.$$

Для того, чтобы определить s_{opt} , воспользуемся OA (20) с переменной структурой. Значит, требуется найти s_{opt} , для которого $s_{opt} = \arg \mathbf{u}(s_{opt}) = \arg \max_{s \in S} \{\mathbf{u}(s)\}$. Пусть OA (20)

находится в состоянии s ; выход I (21) пусть будет $\mathbf{u} = \mathbf{u}(s)$. Учитель должен состояние s по $\mathbf{u} = \mathbf{u}(s)$ оценить следующим образом [34]: если $\mathbf{u}(s) = \max_{s \in S} \{\mathbf{u}(s)\}$, то $y = 0$; иначе $y = 1$.

Для того, чтобы учитель смог оценить значение $\mathbf{u}(s)$, должен априорно располагать ожидаемым значением $\hat{\mathbf{u}} = \mathbf{u}(s) = \mathbf{u} p(\mathbf{u} | s)$ [34]. Но учитель, не располагая априорными данными о случайных неявных возмущениях, не в состоянии определить $\max_{s \in S} \{\mathbf{u}(s)\}$ при известных полезностях состоянческих переходов в (2)/(3). Однако учитель располагает локальными наблюдениями оценок $\hat{\mathbf{u}}$ в предшествующие моменты времени $\hat{\mathbf{u}}_H, \hat{\mathbf{u}}_1, \dots, \hat{\mathbf{u}}_k$, вырабатываемых по полезностям состоянческих переходов в (2)/(3) $\hat{\mathbf{u}} = p(s) \mathbf{u}(\delta(s, x, s))$. Очевидно, что сначала будет мотивировка, создаваемая учителем, очень неточной. С возрастающим количеством выполнений, собственной траектории становятся оценки $\hat{\mathbf{u}}$ все точнее, т.е. будет падать количество некорректно присвоенных нештрафов и штрафов; отсюда название учителя несовершенным.

Несовершенный учитель моделируется упорядоченной тройкой

$$H = \langle \mathbf{R}_0^+, Y, C \rangle,$$

где $C : \{1, 2, \dots, |S|\} \rightarrow [0, 1] : s_i \mapsto p(y = 1 | s_i)$. Так как $\mathbf{u}(s)$ не известна, ограничившись стационарной P – моделью объекта [34], т.е., ограничившись постоянными вероятностями $p(y = 1 | s_i)$ ($i = 1, 2, \dots, |S|$), пишут для поощрения [36]