



ИТОГИ НАУКИ И ТЕХНИКИ

ТЕХНИЧЕСКАЯ
КИБЕРНЕТИКА

Том 29



ГОСУДАРСТВЕННЫЙ КОМИТЕТ СССР
ПО НАУКЕ И ТЕХНИКЕ

АКАДЕМИЯ НАУК СССР

ВСЕСОЮЗНЫЙ ИНСТИТУТ НАУЧНОЙ И ТЕХНИЧЕСКОЙ ИНФОРМАЦИИ
(ВИНИТИ)

ИТОГИ НАУКИ И ТЕХНИКИ

СЕРИЯ ТЕХНИЧЕСКАЯ КИБЕРНЕТИКА

Том 29

МЕТОДЫ УПРАВЛЕНИЯ И ПЛАНИРОВАНИЯ
СЛОЖНЫХ СИСТЕМ

Под редакцией
лауреата Ленинской и Государственной премий
академика С. В. Емельянова

Серия издается с 1967 г.



МОСКВА 1990

Главный редактор информационных изданий ВИНТИ
профессор *П. В. Нестеров*

РЕДАКЦИОННАЯ КОЛЛЕГИЯ

информационных изданий
по автоматике и вычислительной технике

Главный редактор — акад. *С. В. Емельянов*

Члены редакционной коллегии: д. т. н. *А. К. Айламазян*,

д. т. н. *И. А. Болошин*, к. т. н. *Е. В. Гамсахурдия*,

д. т. н. *З. Б. Голембо*, к. т. н. *Е. Б. Дудин* (зам. главного редактора),

к. т. н. *Н. Ф. Зюзев* (ученый секретарь редколлегии),

к. т. н. *Э. М. Квартиркин*, д. т. н. *И. М. Коган*, акад. *В. А. Котельников*,

к. т. н. *В. А. Марасанов*, к. т. н. *Н. Н. Миловидов*, д. т. н. *В. И. Нейман*,

акад. *Г. С. Поспелов*, д. т. н. *А. В. Соколов*, к. т. н. *Ю. Н. Сорокин*,

чл.-корр. АН СССР *Л. Н. Сумароков*, чл.-корр. АН СССР *Я. З. Цыпкин*,

д. т. н. *Н. И. Чистяков*, д. т. н. *Ю. М. Шашков*, д. т. н. *А. В. Шмид*

Научные редакторы: проф. *З. Б. Голембо*, к. т. н. *Н. Н. Миловидов*

Рецензенты: д. ф.-м. н. *А. И. Пропой*, д. т. н. *А. А. Дорофеев*,

д. ф.-м. н. *В. И. Опойцев*

111111 1090

ОПТИМАЛЬНЫЕ МЕТОДЫ АДАПТИВНОЙ ИДЕНТИФИКАЦИИ

Я. З. Цыпкин, А. С. Позняк, С. Н. Тихонов

ВВЕДЕНИЕ

Задача идентификации динамических объектов в режиме их нормальной работы, т. е. задача адаптивной идентификации, состоит в определении структуры и параметров идентифицируемой системы по измерениям наблюдаемых выходных величин и входных воздействий. Для линейных динамических объектов, описываемых линейными разностными уравнениями известного порядка, но с неизвестными коэффициентами, идентификация сводится к оценке этих коэффициентов. Во всех известных методах адаптивной идентификации фигурирует настраиваемая модель, которая по наблюдаемым данным формирует оценку выходной величины объекта, т. е. осуществляет предсказание выходной величины на один шаг вперед. Поэтому такая настраиваемая модель представляет собой предсказатель (предиктор). Параметры предиктора изменяются тем или иным алгоритмом идентификации так, чтобы критерий качества идентификации — функционал невязки, т. е. разности выходных величин объекта и предиктора стремился к минимальному значению. По полученным при этом значениям параметров предиктора определяются непосредственно или при помощи вычислительного устройства оценки параметров объекта.

К настоящему времени опубликовано большое число книг и статей, в которых для идентификации однотипных объектов используются различные структуры предикторов, алгоритмов, а отчасти критериев [1—10]. Выбор того или иного метода идентификации часто определяется вкусами и возможностями исследователя. Появились статьи по сравнению различных методов идентификации [11—13]. Однако до последнего времени не обсуждалась возможность установления оптимальных методов идентификации.

В настоящей работе рассматривается задача обоснования наилучших, оптимальных методов идентификации. Знание оптимальных методов идентификации позволяет, с одной сторо-

ны, достигать предельных возможностей, а с другой стороны, оценивать, насколько те или иные известные методы идентификации далеки от этих предельных возможностей.

Оптимальные методы идентификации основаны на учете имеющейся априорной информации об идентифицируемом объекте и помехе. Они дают возможность при заданном уровне априорной информации однозначно определить основные элементы структурной схемы: оптимальные предикторы, оптимальные алгоритмы и оптимальные критерии качества идентификации. Наличие оптимального метода идентификации делает излишним сравнение выбранных тем или иным путем методов и позволяет сознательно формировать методы идентификации для различных условий их применения. Такие оптимальные методы идентификации были изложены в [14—17]. Здесь приводится несколько иное изложение и обобщение этих результатов. Для того чтобы наглядно проиллюстрировать идею оптимизации методов идентификации, мы не будем останавливаться на многочисленных подробностях, связанных с детальным выводом и формальным обоснованием тех или иных положений.

1. ЗАДАЧА ИДЕНТИФИКАЦИИ

Обычно в литературе [1—14] рассматриваются объекты, модели которых описываются линейными разностными уравнениями вида

$$y_n + \sum_{m=1}^N a_m^* y_{n-m} = \sum_{m=0}^{N_1} b_m^* u_{n-m-k} + \xi_n + \sum_{m=1}^{N_2} c_m^* \xi_{n-m}, \quad (1)$$

где u_n — вход, y_n — выход, ξ_n — ненаблюдаемое возмущение, являющееся независимой случайной величиной, такой, что

$$E\{\xi_n\} = 0 \text{ и } E\{\xi_n \xi_{n-m}\} = \begin{cases} 0, & m=0 \\ \sigma^2, & m \neq 0. \end{cases} \quad (2)$$

Модель типа (1) называют ARMAX-моделью (autoregression—moving average—exogenous input model). Пусть ξ_n при всех $n=1, 2, \dots$ имеет одинаковую симметрическую плотность $p_0(\xi) = p_0(-\xi)$. Если $\{u_n\}$ является стационарным случайным процессом, то в режиме нормальной работы (при $n \rightarrow \infty$) процесс $\{y_n\}$ тоже является стационарным.

Уравнение (1) может быть записано в операторной форме:

$$Q(q)y_n = q^{k+1}P_u(q)u_n + P_\xi(q)\xi_n, \quad (3)$$

где

$$Q(q) = 1 + a_1^*q + \dots + a_N^*q^N \quad (4)$$

характеристический полином, а

$$P_u(q) = b_0^* + b_1^*q + \dots + b_{N_1}^*q^{N_1} \quad (5)$$

$$P_\xi(q) = 1 + c_1^*q + \dots + c_{N_2}^*q^{N_2}$$

— полиномы входных и возмущающих воздействий. Здесь q — оператор задержки на один такт, действующий по правилу $q^m z_n = z_{n-m}$.

Уравнение (3) включает в себя

— авторегрессионную часть (AR) $Q(q)y_n$,

— скользящее среднее (MA) $P_\xi(q)\xi_n$,

— регрессионную часть (X) $q^{k+1}P_u(q)u_n$.

Многочлен $P_\xi(q)$ зависит от точки приложения помехи ξ_n : так, если помеха приложена на выходе объекта, то $P_\xi(q) = Q(q)$, если помеха приложена на входе, то $P_\xi(q) = q^{k+1}P_u(q)$. Естественно, существует точка приложения помехи, когда $P_\xi(q) = 1$. Таким образом, в большинстве случаев коэффициенты многочлена $P_\xi(q)$ функционально связаны с коэффициентами многочленов $Q(q)$ и $P_u(q)$.

Задача оценивания коэффициентов полиномов $Q(q)$ и $P_u(q)$ по наблюдениям $\{y_n\}$ и $\{u_n\}$ состоит в построении оценок $\Theta_n = \Theta_n(y^n, u^{n-k})$, сходящихся в некотором вероятностном смысле к вектору истинных параметров модели $\Theta^T = (a_1^*, \dots, a_N^*; b_0^*, \dots, b_{N_1}^*)$.

Для решения этой задачи введем следующие понятия:

— статический предиктор (предсказатель на один такт вперед)

$$\hat{y}_n = - \sum_{m=1}^N a_m y_{n-m} + \sum_{m=0}^{N_1} b_m u_{n-m-k}, \quad (6)$$

— динамический предиктор

$$\hat{y}_n = - \sum_{m=1}^N a_m \hat{y}_{n-m} + \sum_{m=0}^{N_1} b_m u_{n-m-k}, \quad (7)$$

— невязка (отличие выходов модели и предиктора)

$$e_n(\Theta) = y_n - \hat{y}_n. \quad (8)$$

В качестве критерия точности идентификации, т. е. оценивания параметров модели (1), введем функцию средних потерь вида

$$J(\Theta) = \bar{E}\{F[e_n(\Theta)]\}, \quad (9)$$

где оператор $\bar{E}\{\cdot\} = \lim_{n \rightarrow \infty} n^{-1} \sum_{i=1}^n E\{\cdot\}$.

Обычно выбирают $F(\varepsilon) = \varepsilon^2/2$, что соответствует среднеквадратическому критерию. Реже $F(\varepsilon) = |\varepsilon|$, что соответствует среднемодульному критерию.

Вычисляя градиент от средней функции потерь (9) с учетом (8), запишем условие оптимальности оценок параметров предиктора в виде

$$\nabla J(\Theta) = -\bar{E}\{F'[e_n(\Theta)] \nabla_{\Theta} \hat{y}_n\} = 0. \quad (10)$$

Используя адаптивный подход [18] и, в частности, метод стохастической аппроксимации, запишем рекуррентный алгоритм идентификации в виде

$$\Theta_n = \Theta_{n-1} + \Gamma_n F'[\varepsilon_n(\Theta_{n-1})] \nabla_{\Theta} \hat{y}_n. \quad (11)$$

Часто используются упрощенные варианты алгоритма (11), когда вместо оценок градиента $\nabla_{\Theta} \hat{y}_n$ используется некоторый вектор x_n :

$$\Theta_n = \Theta_{n-1} + \Gamma_n F'[\varepsilon_n(\Theta_{n-1})] x_n. \quad (12)$$

В случае квадратичной функции потерь (когда $F'(\varepsilon) = \varepsilon^2/2$) алгоритм (12) переходит в линейный алгоритм

$$\Theta_n = \Theta_{n-1} + \Gamma_n \varepsilon_n(\Theta_{n-1}) x_n. \quad (13)$$

Этот линейный алгоритм применяется и исследуется в подавляющем большинстве работ по адаптивной идентификации. Выбирая различным образом матрицу усиления Γ_n и вектор наблюдений x_n , из алгоритма (13) можно получить алгоритмы метода стохастического градиента [18—22],

- расширенного метода наименьших квадратов [23—27],
метода инструментальных переменных [28—33],
метода эталонной модели [34],

- метода минимума ошибки прогноза [35].

Эти методы описаны также в [1, 3, 5, 7—10].

Выбирая $F(\varepsilon) = |\varepsilon|$ ($F'(\varepsilon) = \text{sign } \varepsilon$) в алгоритме (12), получим релейный алгоритм

$$\Theta_n = \Theta_{n-1} + \Gamma_n x_n \text{sign } \varepsilon_n(\Theta_{n-1}), \quad (14)$$

соответствующий методу наименьших модулей [36].

Алгоритм (12) при достаточно широком классе функций потерь был рассмотрен в [9, 18, 37]. Как правило, до настоящего времени изучалась сходимости, а иногда и скорость сходимости, линейных алгоритмов типа (13) и знаковых — типа (14).

Немного истории. К сожалению, опыт практического использования многих стандартных алгоритмов идентификации показывает, что они требуют слишком много времени для получения приемлемой точности решения и потому становятся иногда неэффективными. Это вызывает необходимость объяснения их порой странного поведения и определения их возможностей вообще. Использование предикторов типа (6) или (7), как было установлено уже давно, приводит к смещенным оценкам в случае, когда $P_{\varepsilon}(q) = 1$, что подтолкнуло к развитию методов устранения этого смещения [30, 38, 39]. Среди них можно выделить метод инструментальных переменных [33]. Все это, естественно, породило проблему выбора оптимального предиктора, который обеспечивал бы не только несмещенность оценок, но обладал бы еще и другими желаемыми свойствами.

Интересно проследить за дискуссией относительно выбора функции потерь, т. е. критерия качества идентификации (9)

1. Выбор самого критерия оптимальности произволен. Выбор всегда представляет собой компромисс между стремлением приблизиться к реальности и стремлением к математической простоте [140].

2. Часто выбор конкретного вида функций потерь представляет собой компромисс между практической полезностью такой функции и простотой статистического решения задачи [41].

3. В любой задаче мы выбираем функцию стоимости (потерь) из двух соображений. Во-первых, желательно, чтобы стоимость служила адекватной мерой степени удовлетворения потребителя. ... Второе соображение при выборе функции стоимости заключается в том, чтобы задаться функцией стоимости, которая приводила бы к разрешимой задаче [42].

4. При выборе критерия оптимальности в основном приходится руководствоваться не столько строгим отображением физических условий работы и назначения системы, сколько удобством и простотой получения аналитических результатов [43].

5. Отсутствие точных сведений приводит к необходимости выбирать функцию потерь некоторым условным образом. При этом математическая простота рассматривается как важное обстоятельство [44].

6. Нет решительно никаких общих соображений для предпочтения одного критерия другому... Критерий квадратичного отклонения... применяется особенно часто потому, что при использовании этим критерием получается, как правило, простая выкладка [43].

7. Обоснование выбора критерия качества и ограничений на систему выходит за рамки излагаемой ниже теории и базируется на человеческом опыте, здравом смысле и интуиции [44].

8. Выбор критерия принципиально не поддается формализации и, следовательно, невозможно предположить набор правил, руководствуясь которым можно было бы легко выбрать критерий для каждой конкретной задачи [45].

9. Вопрос о выборе критерия для сравнения различных систем, имеющих одинаковое назначение, не может быть решен средствами математики. Этот вопрос следует решать с позиций здравого смысла, исходя из условий работы системы и ее назначения [48].

10. К сожалению, выбор критерия оптимальности более или менее субъективен, а получающиеся алгоритмы существенно зависят от этого выбора [1].

11. В прошлом, как и в настоящее время, продолжается полемика по поводу выбора наилучшего критерия ошибки [49].

12. Выбор критерия приближения является трудным и спорным вопросом [50].

13. Возможно, и есть правила для выбора критерия, но есть ли правила для выбора этих правил [51].

Все вышесказанное свидетельствует, что каждый раз нам приходится иметь дело с неким аппроксимирующим критерием, характеризующим степень отличия выходов объекта и предиктора, оценок и истинных значений и т.п. Иные «содержательные» критерии оптимальности, такие, как эффективность, вес, размеры, энергоемкость синтезируемой системы, как правило, не рассматриваются, поскольку их выбор часто мотивируется технической и экономической спецификой. Многокритериальных задач, или, как иногда говорят, задач векторной оптимизации, мы также не будем здесь затрагивать. Что касается алгоритмов идентификации, то вряд ли существует нехватка в их разнообразии, при этом некоторые из них сравниваются по своим свойствам, хотя в большинстве случаев они существуют и исследуются автономно и независимо один от другого.

Возникает два вопроса:

— каким образом выбор функции потерь влияет на свойства алгоритма идентификации;

— существует ли возможность выбора этих функций таким образом, чтобы улучшить свойства алгоритмов идентификации. Может быть, на эти вопросы не было ответов потому, что ранее не было точных их постановок?

Постановка задачи. Итак, рассмотрим динамический объект, описываемый уравнением

$$Q(q)y_n = q^{n+1}P_u(q)u_n + P_\varepsilon(q)\xi_n. \quad (15)$$

Введем обобщенный критерий идентификации — обобщенные средние потери:

$$J(\Theta) = \bar{E}\{F[C^0(q)\varepsilon_n(\Theta)]\}, \quad (16)$$

где $\varepsilon_n(\Theta) = y_n - \hat{y}_n$ — невязка (ошибка прогноза),

$$C^0(q) = 1 + d_1^0q + \dots + d_{N_s}^0q^{N_s} \quad (17)$$

— устойчивый полином, заданный условием задачи, $F[\cdot]$ — функция потерь.

Задача состоит в формировании оптимальных методов идентификации и включает в себя

— синтез оптимальных предикторов,

— выбор оптимальных функций потерь,

— синтез оптимальных алгоритмов настройки параметров предиктора.

Оптимальными предикторами являются модели идентифицируемого объекта, формирующие прогноз его выхода на один такт вперед и минимизирующие обобщенные средние потери. Оптимальными алгоритмами являются такие алгоритмы, которые обладают предельно возможной или гарантированной скоростью сходимости. Говоря об оптимальных методах идентификации, мы всегда будем понимать их относительно, поскольку каждый из этих оптимальных методов

определяется своей априорной информацией об объекте и внешней среде. Формирование оптимальных методов идентификации, таким образом, требует установления структуры оптимальных предикторов, затем введения критерия характеризующего качество рекуррентных алгоритмов идентификации, и, наконец, определение оптимальных параметров этих алгоритмов, таких, как матричный коэффициент усиления и функция потерь, которые в конечном итоге и определяют средние потери.

Оптимальные предикторы. Оптимальным предиктором является линейное разностное уравнение (модель объекта), формирующее последовательность выходов $\{\hat{y}_n\}$, которая минимизирует при соответствующей настройке ее параметров обобщенный критерий идентификации

$$J(\Theta) = E\{F[\varepsilon_n^c]\}, \quad (18)$$

где

$$\varepsilon = C^0(q)\varepsilon_n = C^0(q)(y_n - \hat{y}_n) \quad (19)$$

— обобщенная, или профильтрованная невязка.

Решение проблемы синтеза оптимального предиктора дает следующая теорема.

Теорема 1 (об оптимальном предикторе). Для устойчивого и минимально-фазового (по отношению к шуму) динамического объекта (15) оптимальный предиктор задается уравнением

$$C^0(q)P_\varepsilon(q)\hat{y}_n = P(q)y_{n-1} + P_u(q)u_{n-1}, \quad (20)$$

где полином $P(q)$ размерности $N_5 = \max(N, N_0 + N_2)$ удовлетворяет полиномиальному уравнению

$$C^0(q)P_\varepsilon(q) = Q(q) + qP(q). \quad (21)$$

Для оптимальных предикторов справедливо следующее равенство:

$$C^0(q)\varepsilon_n = C^0(q)(y_n - \hat{y}_n) = \xi_n, \quad (22)$$

при этом условия оптимальности

$$\nabla J = -\bar{E}\{F'[\xi]\nabla\hat{y}_n\} = 0 \quad (23)$$

выполнены для всех симметричных функций потерь ($F(\varepsilon) = F(-\varepsilon)$) и минимум средних потерь равен

$$J^* = \bar{E}\{F[\xi]\}. \quad (24)$$

Доказательство этой теоремы следует непосредственно из вычислений, использующих (15) и (21). В самом деле,

$$C^0(q)P_\varepsilon(q)\varepsilon_n = [Q(q) + qP(q)]y_n - C^0(q)P_\varepsilon(q)\hat{y}_n.$$

Заменяя $Q(q)y_n$ на эквивалентное выражение из (3), получим

$$C^0(q)P_{\xi}(q)\varepsilon_n = [P_u(q)u_{n-k-1} + P(q)y_{n-1} - C^0(q)P_{\xi}(q)\hat{y}_n] + P_{\xi}(q)\xi_n.$$

Приравнявая нулю выражение в квадратных скобках, мы получаем уравнение для оптимального предиктора (20), равенство (22), а, следовательно, и (24). Так как \hat{y}_n (а, значит, и $\nabla\hat{y}_n$) зависит только от наблюдений до момента n , т. е. не зависит от ξ_n , то, следовательно, $F'(\xi_n)$ и $\nabla\hat{y}_n$ — некоррелированы, а значит,

$$E\{F'[\xi]\nabla\hat{y}_n\} = E\{F'[\xi]\}E\{\nabla\hat{y}_n\}. \quad (25)$$

А поскольку производная функции потерь нечетна ($F'(-\xi) = -F'(\xi)$), то

$$E\{F'[\xi]\} = 0, \quad (26)$$

и (23) вытекает из (25) и (26).

Следовательно, можно заключить, что структура оптимального предиктора не зависит от выбора конкретной функции потерь, т. е. не зависит от критерия качества идентификации.

Предиктор, отличающийся от оптимального, не обладает такой инвариантностью, и выбор различных критериев оптимальности может приводить к смещенным оценкам. В связи с этим дальнейший разговор о выборе функции потерь может быть оправдан лишь в случае, когда предиктор синтезирован оптимальным образом. В противном случае сама постановка задачи о выборе оптимальной функции потерь лишена смысла.

В заключение заметим, что минимальное значение средних потерь J^* (24), естественно, зависит от выбора самой функции потерь $F(\cdot)$. Мы будем находить оптимальную функцию потерь из условия возможного минимума величины J^* . Причина, в силу которой это осуществляется, обсуждается ниже.

Основные формы оптимальных предикторов. Уравнения оптимального предиктора (20), (21) могут быть представлены в виде трех основных эквивалентных форм.

Первая форма:

$$Q(q)\hat{y}_n = P(q)\varepsilon_{n-1} + P_u(q)u_{n-k-1} \quad (27)$$

следует из (20), (21) после исключения из них полинома $C^0(q)P_{\xi}(q)$ и учета (8).

Вторая форма: для новых переменных

$$C^0(q)\bar{y}_n = y_n, \quad C^0(q)\bar{u}_n = u_n \quad (28)$$

имеет вид

$$P_{\xi}(q)\hat{y}_n = P(q)\bar{y}_{n-1} + P_u(q)u_{n-k-1}. \quad (29)$$

Третья форма: следует из (24) и (21) после исключе-

ния полинома $qP(q)$

$$P_{\xi}(q)\hat{y}_n = (C^0(q)P_{\xi}(q) - Q(q))\bar{y}_n + P_u(q)\bar{u}_{n-k-1}, \quad (30)$$

или

$$P_{\xi}(q)\hat{y}_n = P_{\xi}(q)y_n - Q(q)\bar{y}_n + P_u(q)\bar{u}_{n-k-1}.$$

Уравнения (27), (29) и (30) и представляют собой три эквивалентные формы оптимальных предикторов. Рассмотрим некоторые их частные свойства.

Оптимальные параметры предиктора в форме (30) в точности соответствуют параметрам идентифицируемого объекта (15). Оптимальные параметры предикторов (27) и (29) лишь частично соответствуют параметрам идентифицируемого объекта (15), и, следовательно, для определения всех параметров объекта требуется добавить еще и уравнение связи (21). Поэтому предиктор в форме (30) называют прямым предиктором, в то время как предикторы в форме (27), (29), называют непрямыми предикторами первого и второго типа, соответственно.

Адаптивные оптимальные предикторы. Предикторы (7), (29) и (30) в действительности определяют только структуру оптимальных предикторов, т. к. их параметры не известны заранее и подлежат определению. Поэтому такие предикторы с подстраиваемыми по ходу процесса параметрами будем далее называть адаптивными оптимальными предикторами.

Преобразуем соотношения (27), (29) и (30) к рекуррентному виду, заменив в нужных местах коэффициенты многочленов на их текущие оценки. Обозначая эти новые многочлены через $\hat{Q}(q)$, $\hat{P}_u(q)$, $\hat{P}_{\xi}(q)$, $\hat{P}(q)$, перепишем (30) с учетом (19) в виде

$$\hat{y}_n = (C^0(q) - 1)\bar{y}_n + (1 - \hat{Q}(q))\bar{y}_n + \hat{P}_u(q)\bar{u}_{n-k-1} - (1 - \hat{P}_{\xi}(q))\varepsilon_n. \quad (31)$$

Обозначим вектор параметров и вектор наблюдений через

$$\Theta_{(0)}^T = (a_1, \dots, a_N, b_0, \dots, b_{N_1}, c_1, \dots, c_{N_2}), \\ x_{(0)}^T = (-\bar{y}_{n-1}, \dots, -\bar{y}_{n-N}, \bar{u}_{n-k-1}, \dots, \bar{u}_{n-k-1-N_1}, \\ \varepsilon_{n-1}, \dots, \varepsilon_{n-N_2}). \quad (32)$$

Тогда уравнение прямого адаптивного предиктора можно кратко записать в виде

$$\hat{y}_n = (C_0(q) - 1)\bar{y}_{n-1} + \Theta_{(0)}^T x_{(0)}^{(0)}. \quad (33)$$

Для непрямого адаптивного предиктора первого типа, соответ-

ствующего (29), по аналогии имеем

$$\hat{y}_n = \hat{P}(q)\bar{y}_{n-1} + \hat{P}_u(q)\bar{u}_{n-k-1} + (1 - \hat{P}_\xi(q))\bar{y}_n, \quad (34)$$

Вводя другие обозначения для вектора параметров и вектора наблюдений

$$\begin{aligned} \Theta_{(1)}^T &= (h_0, \dots, h_{N_s}, b_0, \dots, b_{N_1}, c_1, \dots, c_{N_2}), \\ x_{(1)}^T &= (\bar{y}_{n-1}, \dots, \bar{y}_{n-N_s-1}, \bar{u}_{n-k-1}, \dots, \bar{u}_{n-k-1-N_1}, \\ &\quad -\hat{y}_{n-1}, \dots, -\hat{y}_{n-N_2}), \end{aligned} \quad (35)$$

можно записать уравнение адаптивного предиктора первого типа

$$\hat{y}_n = \Theta_{(1)}^T x_n^{(1)}. \quad (36)$$

И, наконец, для непрямого адаптивного предиктора второго типа, соответствующего (27), получаем

$$\hat{y}_n = (1 - Q(\hat{y}))\hat{y}_n + \hat{P}_u(q)u_{n-k-1} + \hat{P}(q)\varepsilon_{n-1}. \quad (37)$$

При других обозначениях

$$\begin{aligned} \Theta_{(2)}^T &= (a_1, \dots, a_N, b_0, \dots, b_{N_1}, h_0, \dots, h_{N_s}), \\ x_{(2)}^T &= (-\hat{y}_{n-1}, \dots, -\bar{y}_{n-N}, u_{n-k-1}, \dots, u_{n-k-1-N_1}, \\ &\quad \varepsilon_{n-1}, \dots, \varepsilon_{n-N_s}) \end{aligned} \quad (38)$$

можно записать уравнение адаптивного предиктора в форме

$$y_n = \Theta_{(2)}^T x_n^{(2)}. \quad (39)$$

Естественно, к уравнениям не прямых адаптивных предикторов (37) и (39) следует добавить полиномиальное уравнение

$$C^0(q)\hat{P}_\xi(q) = \hat{Q}(q) + q\hat{P}(q). \quad (40)$$

В частном случае при $C^0(q) \equiv 1$ прямой адаптивный предиктор назывался оптимальной настраиваемой моделью [14, 17]. Аналогичные адаптивные предикторы использовались и в методах минимума ошибки прогноза, максимума правдоподобия при нормальных шумах и при псевдолинейной регрессии [9, 52—57].

Непрямые адаптивные предикторы первого и второго типов рассматривались в [57, 58]. Параметры адаптивного предиктора настраивались алгоритмами идентификации таким образом, чтобы либо средние потери с течением времени стремились бы к нулю, либо оценки этих параметров в полном наборе или частично удовлетворяли соответствующим полиномиальным уравнениям.

Рассмотрим обобщенный критерий качества идентификации (18), в котором обобщенная невязка (31) формируется оптимальным предиктором:

$$J(\Theta) = \bar{E}\{F[\varepsilon_n^C(\Theta)]\}, \quad (41)$$

где $\Theta = \Theta_{(k)}$, $k=0, 1, 2$, зависит от формы адаптивного оптимального предиктора.

В этом случае условия оптимальности (10) могут быть записаны в форме

$$\nabla J(\Theta) = -\bar{E}\{F'[\varepsilon_n^C(\Theta)]C^0(q)\nabla_{\Theta}\hat{y}_n\} = 0. \quad (42)$$

Выражение

$$C^0(q)\nabla_{\Theta}\hat{y}_n = v_n \quad (43)$$

задает так называемую функцию чувствительности. Нетрудно выразить эту функцию через имеющиеся к данному моменту наблюдения. Используя (27), (29) и (30), получим

$$v_n = \frac{1}{\hat{S}_{(k)}(q)} x_n^{(k)}, \quad (44)$$

где

$$\hat{S}_{(k)}(q) = \begin{cases} \hat{P}_\xi(q), & \text{при } k=0, 1 \\ \hat{P}(q), & \text{при } k=2 \end{cases}. \quad (45)$$

Заметим, что v_n зависит от настраиваемого полинома $\hat{S}_{(k)}(q)$ и не зависит от известного, задаваемого априори, полинома $C^0(q)$. Уравнение чувствительности (42) можно представить в форме

$$\hat{S}_{(k)}(q)v_n = x_n^{(k)} \quad (46)$$

или в явной рекуррентной форме

$$v_n = (1 - \hat{S}_{(k)}(q))v_n + x_n^{(k)}. \quad (47)$$

С учетом (43) условие оптимальности (42) можно записать в форме

$$\nabla J(\Theta) = -\bar{E}\{F'[\varepsilon_n^C(\Theta)]v_n\} = 0, \quad (48)$$

где $v(n)$ задается уравнениями (46) или (47). Здесь следует напомнить, что условие оптимальности (47) при $\Theta = \Theta^*$ остается справедливым для любых четных функций потерь.

Условие оптимальности (48) порождает рекуррентные алгоритмы вида

$$\Theta_n = \Theta_{n-1} + \Gamma_n F'[\varepsilon_n(\Theta_{n-1})]v_n. \quad (49)$$

Обычно из условий сходимости матрица усиления выбирается в виде

$$\Gamma_n = n^{-1}\Gamma, \quad (50)$$

где $\Gamma > 0$ — постоянная, положительно определенная, диагональная матрица.

Точность любых оценок Θ_n , в том числе в порождаемых рекуррентным алгоритмом, определяется матрицей ковариации ошибки (МКО)

$$V_n = E\{(\Theta_n - \Theta^*)(\Theta_n - \Theta^*)^T\}. \quad (51)$$

Качество рекуррентных алгоритмов (49) будем характеризовать скоростью их сходимости, а именно, нормированной асимптотической матрицей ковариации (АМКО)

$$V = \lim_{n \rightarrow \infty} nV_n = \lim_{n \rightarrow \infty} nE\{(\Theta_n - \Theta^*)(\Theta_n - \Theta^*)^T\}. \quad (52)$$

АМКО удовлетворяет матричному уравнению Ляпунова [15, 17]

$$GV + VG = \Gamma \nabla^2 J(\Theta^*) \Gamma, \quad (53)$$

где

$$G = 1/2I - \Gamma \nabla^2 J(\Theta^*), \quad (54)$$

а I — единичная матрица.

Входящая в (53), (54) матрица Гесса $\nabla^2 J(\Theta^*)$ может быть представлена в виде

$$\nabla^2 J(\Theta^*) = -E\{F''[\xi]\}A(\Theta^*, \sigma^2(p_0)) > 0. \quad (55)$$

Здесь

$$A(\Theta^*, \sigma^2(p_0)) = E\{v_n v_n^T\} \quad (56)$$

— нормированная информационная матрица (НИМ), зависящая, вообще говоря, от Θ^* и дисперсии помех $\sigma^2(p_0)$. Структура НИМ такова:

$$A(\Theta^*, \sigma^2(p_0)) = A_I(\Theta^*) + \sigma^2(p_0)A_{II}(\Theta^*), \quad (57)$$

где $A_I(\Theta^*) \geq 0$ и $A_{II}(\Theta^*) \geq 0$ — неотрицательно определенные симметричные матрицы.

Задаваясь матрицей усиления Γ и решая уравнение (53) относительно V , получим

$$V = V(\Gamma) = E\{\varphi^2(\xi)\} \int_0^\infty e^{Gt} \Gamma \nabla^2 J(\Theta^*) \Gamma^T e^{G^T t} dt.$$

Эта матрица и характеризует качество алгоритма (49) при данном Γ .

Кроме того, можно показать [37, 59], что в соответствии с информационным неравенством Крамера—Рао имеет место матричное неравенство

$$\lim_{n \rightarrow \infty} nV_n \geq [I_F(p_0) A(\Theta^*, \sigma^2(p_0))]^{-1}. \quad (58)$$

Отсюда следует, что АМКО удовлетворяет неравенству

$$V \geq [I_F(p_0) A(\Theta^*, \sigma^2(p_0))]^{-1}. \quad (59)$$

В (58) и (59) фигурирует фишеровская информация [59]:

$$I_F(p_0) = E\left\{\left(\frac{p_0'(\xi)}{p_0(\xi)}\right)^2\right\}. \quad (60)$$

Заметим, что матричные неравенства (57), (58) означают, что разность левой и правой матриц представляет собой неотрицательно-определенную матрицу. Правая часть неравенства (58) зависит только от характеристик помех (плотности распределения и дисперсии) и параметров объекта и определяет предельно возможную нижнюю границу АМКО, т. е. предельно возможную скорость сходимости любых оценок, в том числе и оценок, порождаемых рекуррентными алгоритмами.

В ряде случаев мерой качества рекуррентных алгоритмов вместо АМКО (42) может служить асимптотическое уклонение (АУ)

$$w = \lim_{n \rightarrow \infty} n(E\{J_2(\Theta_n)\} - J_2^*), \quad (61)$$

где $J_2(\Theta_n) = E\{\varepsilon_n^2(\Theta_n)\}$ — квадратичные средние потери, J_2^* — минимальное их значение.

Разлагая $J_2(\Theta_n)$ в ряд Тейлора, получим приближение

$$J_2(\Theta_n) \approx J_2^* + \frac{1}{2}(\Theta_n - \Theta^*)^T \nabla^2 J_2(\Theta^*)(\Theta_n - \Theta^*). \quad (62)$$

Но в этом случае $F''[\xi] = 2$ и, значит,

$$\nabla^2 J_2(\Theta^*) = 2A(\Theta^*, \sigma^2(p_0)).$$

Следовательно,

$$J_2(\Theta_n) - J_2^* \approx (\Theta_n - \Theta^*)^T A(\Theta^*, \sigma^2(p_0))(\Theta_n - \Theta^*). \quad (63)$$

Поскольку

$$(\Theta_n - \Theta^*)^T A(\Theta^*, \sigma^2(p_0))(\Theta_n - \Theta^*) = \text{tr}[A(\Theta^*, \sigma^2(p_0))V_n],$$

имеем

$$J_2(\Theta_n) - J_2^* \approx \text{tr}[A(\Theta^*, \sigma^2(p_0))V_n]. \quad (64)$$

Умножая обе части (62) на n и переходя к пределу на основании (52), получаем

$$w = \text{tr}[A(\Theta^*, \sigma^2(p_0))V], \quad (65)$$

где V находится из матричного уравнения (53), (54). Таким образом, АУ определяется следом произведения НИМ и АМКО. АМКО характеризует аргументную скорость сходимости, тогда как АУ характеризует критериальную скорость сходимости. В невырожденном случае $V > 0$ чем выше аргументная скорость сходимости, тем выше и критериальная скорость сходимости.

Оптимизация алгоритмов идентификации. Качество алгоритмов идентификации

$$\Theta_n = \Theta_{n-1} + n^{-1} \Gamma F'[\varepsilon_n^c(\Theta_{n-1})] v_n \quad (66)$$

зависит от матрицы $\Gamma > 0$ и нелинейной функции $F'[e]$ — производной функции потерь. Поэтому целесообразно их выбирать так, чтобы скорость сходимости алгоритма была максимальной, т. е. чтобы АМКО были минимальны.

Рассмотрим вначале оптимизацию алгоритма (66) по матрице усиления Γ .

Теорема 2 (об оптимальной матрице усиления Γ_0). Оптимальная матрица усиления равна

$$\Gamma = \Gamma_0 = A^{-1}(\Theta^*, \sigma^2(p_0)) \frac{1}{E\{F''(\xi)\}}. \quad (67)$$

При этом АМКО и АУ минимальны по Γ и равны соответственно

$$V_{\min} = V(F', p_0) = P(F', p_0) A^{-1}(\Theta^*, \sigma^2(p_0)), \quad (68)$$

$$W_{\min} = P(F', p_0) N_0, \quad N_0 = \text{rank } A(\Theta^*, \sigma^2(p_0)), \quad (69)$$

где

$$\rho(F', p_0) = \frac{E\{F'^2[\xi]\}}{E\{F''[\xi]\}}. \quad (70)$$

Доказательство следует из условия оптимальности

$$\left. \frac{\partial V(\Gamma_0 + \lambda \delta \Gamma_0)}{\partial \lambda} \right|_{\lambda=0} = 0, \quad (71)$$

найденного в силу матричного уравнения (53), которое приводится к виду

$$E\{F''[\xi]\} A(\Theta^*, \sigma^2(p_0)) \Gamma_0 = I,$$

откуда следует (67). Подставляя Γ_0 из (67) и $\nabla^2 J(\Theta^*)$ из (55) в матричное уравнение относительно V (53), (54) и в выражение для w (65), получаем минимальные по Γ АМКО (68), (70) и АУ (69), (70). Заменяя в алгоритме идентификации (66) матрицу усиления Γ на оптимальную — Γ_0 (67) получаем оптимальный по матрице усиления алгоритм

$$\begin{aligned} \Theta_n &= \Theta_{n-1} + \Gamma_{0,n} F' [e_n(\Theta_{n-1})] v_n, \\ v_n &= (I - \hat{S}_{(k)}(q)) v_n + x_n^{(k)}, \end{aligned} \quad (72)$$

$$\Gamma_{0,n} = n^{-1} \frac{1}{E\{F''(\xi)\}} A^{-1}(\Theta^*, \sigma^2(p_0)).$$

Скалярный множитель $P(F', p_0)$, входящий в АМКО V_0 (68) и АУ w_0 (69), зависит от производной функции потерь $F'[e]$ и плотности распределения помех $p_0(\xi)$. Это дает основание выбрать $F'[e]$, а, значит, и функцию потерь $F[e]$ так, чтобы минимизировать АМКО и АУ. Оптимальная производная функции потерь определяется как

$$F'[e] = F_0'[e] = \arg \min_{F'} \rho(F', p_0). \quad (73)$$

Алгоритмы, оптимальные как по матрице усиления, так и по производной функции потерь, можно назвать абсолютно оптимальными алгоритмами.

Теорема 3 (об оптимальной функции потерь и ее производной). Оптимальная функция потерь и ее производная равны

$$F_0[e] = -\ln p_0(\xi) \Big|_{\xi=e}, \quad (74)$$

$$F_0'[e] = -\frac{p_0'(\xi)}{p_0(\xi)} \Big|_{\xi=e} \quad (75)$$

при этом

$$P_{\min} = P(F_0', p_0) = I_F^{-1}(p_0), \quad (76)$$

где

$$I_F(p_0) = E\{F_0'^2[\xi]\} = E\{F_0''[\xi]\} = E\left\{\left(\frac{p_0'(\xi)}{p_0(\xi)}\right)^2\right\} \quad (77)$$

— фишеровская информация (60), и, значит,

$$V(F', p_0) \geq V(F_0', p_0) = V(p_0), \quad (78)$$

где

$$V(p) = \frac{1}{I_F(p_0)} A^{-1}(\Theta^*, \sigma^2(p_0)) \quad (79)$$

— минимально возможная АМКО и

$$w(F', p_0) \geq w(F_0', p_0) = w(p_0), \quad (80)$$

где

$$w(p_0) = \frac{N_c}{I_F(p_0)} \text{ и } N_c = \text{rank } A(\Theta^*, \sigma^2(p_0)). \quad (81)$$

Доказательство следует из условия оптимальности

$$\left. \frac{\partial p_\lambda(F_0' + \lambda \delta F', p_0)}{\partial \lambda} \right|_{\lambda=0} = 0,$$

которое выполняется, когда

$$\frac{\int_{-\infty}^{\infty} p_0(\xi) F_0'[\xi] d\xi}{\int_{-\infty}^{\infty} p_0(\xi) F_0'^2[\xi] d\xi} = \frac{p_0'(\xi)}{\int_{-\infty}^{\infty} p_0'(\xi) F_0'[\xi] d\xi}.$$

Умножая обе части этого равенства на $F_0'[\xi]$, получаем

$$\frac{p_0(\xi) F_0'^2[\xi]}{\int_{-\infty}^{\infty} p_0(\xi) F_0'^2[\xi] d\xi} = \frac{p_0'(\xi) F_0'[\xi]}{\int_{-\infty}^{\infty} p_0'(\xi) F_0'[\xi] d\xi}.$$

Но равенство таких нормированных функций имеет место лишь тогда, когда сами функции пропорциональны, т. е.

$$p_0(\xi) F_0'^2[\xi] = k_1 p_0'(\xi) F_0'[\xi],$$

где k_1 — коэффициент пропорциональности.

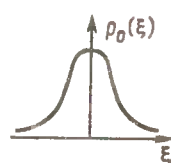
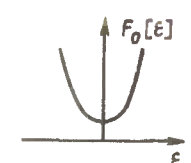
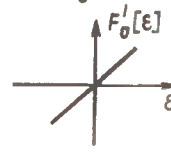
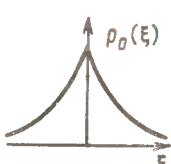
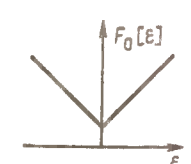
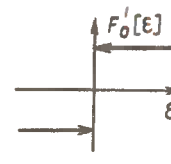
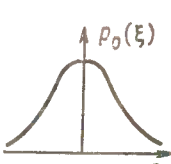
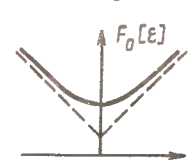
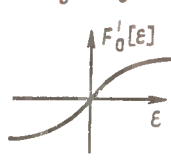
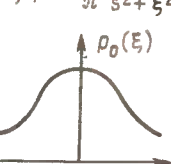
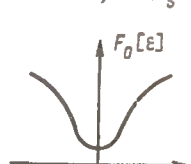
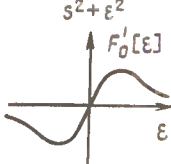
Отсюда следует, что

$$F_0'[k] = k_1 \frac{p_0'(\xi)}{p_0(\xi)}$$

и значит,

$$F_0[\xi] = k_1 \ln p_0(\xi) + k_2.$$

Таблица 1

№	$p_0(\xi)$	$F_0[\varepsilon] = -\ln p_0(\xi) _{\xi=\varepsilon}$	$F_0'[\varepsilon] = -\frac{p_0'(\xi)}{p_0(\xi)} _{\xi=\varepsilon}$
1	$N(0, \sigma^2) \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{\xi^2}{2\sigma^2}}$ 	$\frac{\xi^2}{2\sigma^2} + \ln \sqrt{2\pi} \sigma$ 	$\frac{\xi}{\sigma^2}$ 
2	$L(0, s) \frac{1}{2s} e^{-\frac{\xi}{s}}$ 	$\frac{ \xi }{s} + \ln 2s$ 	$\frac{1}{s} \text{sign } \varepsilon$ 
3	$se(0, s) \frac{1}{2s} \text{sech}^2 \frac{\xi}{s}$ 	$2 \ln \text{sech} \frac{\xi}{s} + \ln 2s$ 	$\frac{2}{s} \text{th} \frac{\xi}{s}$ 
4	$C(0, s) \frac{s}{\pi} \frac{1}{s^2 + \xi^2}$ 	$\ln(s^2 + \varepsilon^2) + \ln \frac{\pi}{s}$ 	$\frac{2\varepsilon}{s^2 + \varepsilon^2}$ 

Из простых геометрических соображений следует, что параметр k_1 (определяющий основание логарифма) должен быть отрицательным $k_1 < 0$, что же касается параметра k_2 , то он определяет минимальное значение $F_0[0]$. При $k_2 > -k_1 \ln p_0(\xi)$ всегда $F_0[\xi] > 0$. Оптимальная функция потерь не единственна, а зависит от параметров k_1, k_2 . Этот факт является следствием того, что АМКО и АУ зависят не от $F_0[\xi]$, а от $F_0'[\xi]$. Для задач идентификации конкретные значения этих параметров несущественны. Удобно принять $k_1 = -1$, а $k_2 = 0$. Таким образом, оптимальная функция потерь равна логарифму плотности распределения помех с обратным знаком

$$F_0[\varepsilon] = -\ln p_0(\xi) |_{\xi=\varepsilon},$$

а ее производная — отношению производной плотности распределения помех к самой плотности распределения помех с обратным знаком, или негинформанту

$$F_0'[\varepsilon] = \frac{p_0'(\xi)}{p_0(\xi)} |_{\xi=\varepsilon}.$$

Подставляя $F_0'[\varepsilon]$ в (70), убеждаемся в справедливости (76) — (78). Поскольку $V(p_0)$ совпадает с нижней границей неравенства Крамера—Рао (59), то оптимальной функции потерь соответствует предельно возможная аргументная скорость сходимости. Критериальная скорость сходимости, как видно из (81), также является предельно возможной. Примеры оптимальных функций потерь и их производных приведены в табл. 1.

Для нормальной (гауссовой) плотности распределения оптимальная функция потерь квадратична, а ее производная линейна. Для экспоненциальной (лапласовой) плотности распределения функция потерь модульная, а ее производная релейна. Отметим еще один интересный факт относительно оптимальной функции потерь. Для семейства функций потерь

$$F[\varepsilon] = -\ln p(\xi) |_{\xi=\varepsilon},$$

где $p(\xi)$ — произвольная симметричная плотность распределения, удовлетворяется неравенство [58]

$$E \left\{ \ln \frac{p_0(\xi)}{p(\xi)} \right\} \geq 0, \quad (82)$$

откуда следует

$$-E \{ \ln p_0(\xi) \} \leq -E \{ \ln p(\xi) \}. \quad (83)$$

Это значит, что для оптимальной функции потерь средние потери достигают минимума миниморума. Разумеется, все это справедливо для оптимального предиктора.

Заменяя в оптимальных по матрице усиления алгоритмах $F'[\varepsilon]$ на $F_0'[\varepsilon]$ и $F''[\varepsilon]$ на $F_0''[\varepsilon]$, получаем абсолютно оптимальные алгоритмы

$$\Theta_n = \Theta_{n-1} + \Gamma_n F_0'[\varepsilon_n(\Theta_{n-1})] \mathbf{v}_n, \quad (84)$$

$$\mathbf{v}_n = (1 - S_{(h)}(q)) \mathbf{v}_n + \mathbf{x}_n^{(h)} \quad (85)$$

При этом оптимальная матрица Γ_n равна

$$\Gamma_n = \Gamma_{0,n} = n^{-1} [E \{ F_0'[\xi] \} A(\Theta^*, \sigma^2(p_0))]^{-1}, \quad (86)$$

или, принимая во внимание (77),

$$\Gamma_{0,n} = n^{-1} [I_F(p_0) A(\Theta^*, \sigma^2(p_0))]^{-1}. \quad (87)$$

Вспоминая определение НИМ $A(\Theta^*, \sigma^2(p_0))$ (57) и учитывая,

что $v(n)$ и $\xi(n)$ не коррелированы, запишем $\Gamma_{0,n}$ еще и в такой форме

$$\Gamma_{0,n} = n^{-1} [E\{F''[\xi] v_n v_n^T\}]^{-1}. \quad (88)$$

Абсолютно оптимальные алгоритмы обладают предельно возможной скоростью сходимости. Для них АМКО (80) совпадает с нижней границей неравенства Крамера—Рао (59).

Таким образом, учет априорной информации о помехах позволяет определить абсолютно оптимальные алгоритмы.

Реализуемые абсолютно оптимальные алгоритмы. Непосредственное использование абсолютно оптимальных алгоритмов наталкивается на затруднения, связанные с вычислением матрицы усиления Γ_n . Как видно из (81)—(83), для их вычисления необходимо либо знание неизвестного вектора параметров объекта Θ^* (см. (84), (86)), либо знание статистических свойств вектора чувствительности v_n (88), а значит, и вектора наблюдений. Для обхода этих затруднений найдем рекуррентные уравнения оценок матрицы усиления использующие лишь наблюдения. Рассмотрим обратную к (88) оптимальную матрицу усиления $\Gamma_{0,n}^{-1}$

$$\Gamma_{0,n}^{-1} = nE\{F''[\xi] v_n\}. \quad (89)$$

Заменим ее эмпирической оценкой

$$\hat{\Gamma}_{0,n}^{-1} = \sum_{m=1}^n F''[\xi_m] v_m v_m^T. \quad (90)$$

Отсюда получаем рекуррентное уравнение

$$\hat{\Gamma}_{0,n}^{-1} = \hat{\Gamma}_{0,n-1}^{-1} + F''[\xi_n] v_n v_n^T. \quad (91)$$

Но помеху ξ_n можно оценивать невязкой (см. (23)). Поэтому окончательно рекуррентные уравнения относительно матриц усиления запишутся в виде

$$\hat{\Gamma}_0^{-1} = \hat{\Gamma}_{0,n-1}^{-1} + F''[\varepsilon_n^C(\Theta_{n-1})] v_n v_n^T. \quad (92)$$

Пользуясь формулой обращения [59, 5, 9], получаем рекуррентное уравнение относительно матрицы усиления

$$\hat{\Gamma}_{0,n} = \hat{\Gamma}_{0,n-1} - \hat{\Gamma}_{0,n-1} v_n [(F_0''[\xi_n])^{-1} + v_n^T \hat{\Gamma}_{0,n-1} v_n]^{-1} v_n^T \hat{\Gamma}_{0,n-1}, \quad (93)$$

которое и определяет оценку $\hat{\Gamma}_{0,n}$ по предшествующей оценке $\hat{\Gamma}_{0,n-1}$ и v_{n-1} .

Далее для простоты записи реализуемых абсолютно оптимальных алгоритмов мы вместо уравнения (93) будем выписывать уравнение (92), хотя при практическом применении алгоритмов следует пользоваться уравнением (93). Таким образом, реализуемые абсолютно оптимальные алгоритмы могут быть

представлены в виде

$$\begin{aligned} \Theta_n &= \Theta_{n-1} + \hat{\Gamma}_{0,n} F_0' [\varepsilon_n^C(\Theta_{n-1})] v_n, \\ \hat{\Gamma}_{0,n}^{-1} &= \hat{\Gamma}_{0,n-1}^{-1} + F_0'' [\varepsilon_n^C(\Theta_{n-1})] v_n v_n^T, \\ v_n &= (1 - \hat{S}_{(k)}(q)) v_n + x_n^{(k)}. \end{aligned} \quad (94)$$

Эти алгоритмы следует дополнить начальными значениями

$$\Theta_0 = \Theta^0, \Gamma_0 = \gamma I, v_0 = 0, \quad (95)$$

где Θ^0, γ — постоянные. Обычно $\gamma \gg 1$.

Конкретный вид абсолютно оптимальных алгоритмов зависит от формы адаптивного предиктора. Так, используя обозначение (45) и уравнения адаптивного предиктора (прямого (32), (33), непрямого первого рода (35), (36) и непрямого второго рода (38), (39)), получаем из общих алгоритмов (94), дополнив их выражениями для невязок, основные типы алгоритмов: прямые алгоритмы

$$\begin{aligned} \Theta_n^{(0)} &= \Theta_{n-1}^{(0)} + \hat{\Gamma}_{0,n} F_0' [\varepsilon_n^C(\Theta_{n-1}^{(0)})] v_n, \\ \hat{\Gamma}_0^{-1}(n) &= \hat{\Gamma}_0^{-1}(n-1) + F_0'' [\varepsilon_n^C(\Theta_{n-1}^{(0)})] v_n v_n^T, \\ \varepsilon_n^C(\Theta_{n-1}^{(0)}) &= y(n) - (C^0(q) - 1) \bar{y}(n) + \Theta_{n-1}^{(0)T} x_n^{(0)}, \\ v_n &= (1 - \hat{P}_{\xi}(q)) v_n + x_n^{(0)}; \end{aligned} \quad (96)$$

непрямые алгоритмы первого рода

$$\begin{aligned} \Theta_n^{(1)} &= \Theta_{n-1}^{(1)} + \hat{\Gamma}_{0,n} F_0' [\varepsilon_n^C(\Theta_{n-1}^{(1)})] v_n, \\ \hat{\Gamma}_0^{-1}(n) &= \hat{\Gamma}_0^{-1}(n-1) + F_0'' [\varepsilon_n^C(\Theta_{n-1}^{(1)})] v_n v_n^T, \\ \varepsilon_n^C(\Theta_{n-1}^{(1)}) &= y(n) - \Theta_{n-1}^{(1)T} x_n^{(1)}, \\ v_n &= (1 - \hat{P}_{\xi}(q)) v_n + x_n^{(1)}; \end{aligned} \quad (97)$$

непрямые алгоритмы второго рода

$$\begin{aligned} \Theta_n^{(2)} &= \Theta_{n-1}^{(2)} + \hat{\Gamma}_{0,n} F_0' [\varepsilon_n^C(\Theta_{n-1}^{(2)})] v_n, \\ \hat{\Gamma}_0^{-1}(n) &= \hat{\Gamma}_0^{-1}(n-1) + F_0'' [\varepsilon_n^C(\Theta_{n-1}^{(2)})] v_n v_n^T, \\ \varepsilon_n^C(\Theta_{n-1}^{(2)}) &= y(n) - \Theta_{n-1}^{(2)T} x_n^{(2)}, \\ v_n &= (1 - \hat{P}(q)) v_n + x_n^{(2)}. \end{aligned} \quad (98)$$

В общем случае эти алгоритмы нелинейны относительно преобразованной невязки ε^c . Если плотность распределения помехи нормальна, то, как следует из табл. 1,

$$F'[\varepsilon^c] = \varepsilon^c, F''[\varepsilon^c] = 1. \quad (99)$$

В этом случае из (96)—(98) получаем линейные относительно преобразованной невязки ε^c алгоритмы основных типов:

прямые алгоритмы

$$\begin{aligned}\Theta_n^{(0)} &= \Theta_{n-1}^{(0)} + \hat{\Gamma}_{0,n} \varepsilon_n^C (\Theta_{n-1}^{(0)}) \mathbf{v}_n, \\ \hat{\Gamma}_0^{-1}(n) &= \hat{\Gamma}_0^{-1}(n-1) + \mathbf{v}_n \mathbf{v}_n^T, \\ \varepsilon_n^C (\Theta_{n-1}^{(0)}) &= y(n) - (C^0(q) - 1) \bar{y}(n) + \Theta_{n-1}^{(0)T} \mathbf{x}_n^{(0)}, \\ \mathbf{v}_n &= (1 - \hat{P}_{\xi}(q)) \mathbf{v}_n + \mathbf{x}_n^{(0)};\end{aligned}\quad (100)$$

непрямые алгоритмы первого рода

$$\begin{aligned}\Theta_n^{(1)} &= \Theta_{n-1}^{(1)} + \hat{\Gamma}_{0,n} \varepsilon_n^C (\Theta_{n-1}^{(1)}) \mathbf{v}_n, \\ \hat{\Gamma}_0^{-1}(n) &= \hat{\Gamma}_0^{-1}(n-1) + \mathbf{v}_n \mathbf{v}_n^T, \\ \varepsilon_n^C (\Theta_{n-1}^{(1)}) &= y(n) - \Theta_{n-1}^{(1)T} \mathbf{x}_n^{(1)}, \\ \mathbf{v}_n &= (1 - \hat{P}_{\xi}(q)) \mathbf{v}_n + \mathbf{x}_n^{(1)};\end{aligned}\quad (101)$$

непрямые алгоритмы второго рода

$$\begin{aligned}\Theta_n^{(2)} &= \Theta_{n-1}^{(2)} + \hat{\Gamma}_{0,n} \varepsilon_n^C (\Theta_{n-1}^{(2)}) \mathbf{v}_n, \\ \hat{\Gamma}_0^{-1}(n) &= \hat{\Gamma}_0^{-1}(n-1) + \mathbf{v}_n \mathbf{v}_n^T, \\ \varepsilon_n^C (\Theta_{n-1}^{(2)}) &= y(n) - \Theta_{n-1}^{(2)T} \mathbf{x}_n^{(2)}, \\ \mathbf{v}_n &= (1 - \hat{P}(q)) \mathbf{v}_n + \mathbf{x}_n^{(2)}.\end{aligned}\quad (102)$$

Для простого квадратического критерия качества идентификации $J = E \{ \varepsilon_n^2(\Theta) \}$ в приведенных алгоритмах следует положить $C^0(q) \equiv 1$. В этом случае $\varepsilon_n^C(\Theta) = \varepsilon_n(\Theta)$, в выражении невязки прямых алгоритмов (96), (100) устраняется первое слагаемое. Кроме того, во всех алгоритмах отпадает необходимость в фильтрации наблюдений.

Оптимизация на классе алгоритмов идентификации. В тех случаях, когда априорная информация о помехах неполна, их плотность распределения $p_0(\xi)$ неизвестна. Поэтому мы не можем определить оптимальную функцию потерь $F_0[\varepsilon]$ (74) и, следовательно, воспользоваться абсолютно оптимальными алгоритмами (94). Однако кое-какие сведения о плотности распределения $p_0(\xi)$ нам могут быть известны. Например, может быть известно, что плотность распределения «приближенно нормальна», или что дисперсия помехи ограничена каким-то числом, или что плотность распределения финитна и т. д. Объем этой информации может быть различным: в одном предельном случае плотность распределения полностью известна, в другом — о ней известно лишь то, что она существует. Так или иначе, можно считать, что известен некоторый класс плотностей распределения P , которому принадлежит истинная плотность распределения $p_0(\xi)$. Приведем несколько примеров классов плотностей распределения [15].

1. Класс невырожденных плотностей распределения P_1

$$P_1 = \left\{ p_0(\xi) : p_0(0) \geq \frac{1}{2s} > 0 \right\}. \quad (103)$$

Это наиболее широкий класс.

2. Класс плотностей распределения с ограниченной дисперсией P_2

$$P_2 = \left\{ p_0(\xi) : \int_{-\infty}^{\infty} \xi^2 p_0(\xi) d\xi = \delta^2(p_0) \leq \delta_1^2 \right\}. \quad (104)$$

3. Класс приближенно нормальных плотностей распределения P_3

$$P_3 = \{ p_0(\xi) : p_0(\xi) \geq (1-\alpha) p_N(\xi) \}, \quad (105)$$

где $0 \leq \alpha < 1$ и $p_N(\xi) = N(0, \sigma^2)$ — нормальная плотность распределения.

4. Класс приближенно равномерных плотностей распределения P_4

$$P_4 = \{ p_0(\xi) : p_0(\xi) \geq (1-\alpha) p_R(\xi) \}, \quad (106)$$

где $0 \leq \alpha < 1$ и $p_R(\xi) = R(0, 1)$ — равномерная плотность распределения, равна нулю при $|\xi| > 1$. При этом всегда предполагается, что $p_0(\xi) = p_0(-\xi)$ и $-\infty \int p_0(\xi) d\xi = 1$. Возможны, разумеется, и иные классы, но для иллюстрации достаточно приведенных.

Возникает вопрос: как определить оптимальную функцию потерь в этих условиях?

Для этой цели необходимо ввести понятие оптимальности на классе [17, 18]. Разумно поступить следующим образом: найти в рассматриваемом классе «наименее благоприятную» плотность распределения $p_*(\xi)$, которая максимизирует АМКО (77) либо АУ, (78) и по ней определить оптимальную на классе функцию потерь.

В отличие от оптимальной функции потерь, которая одинакова при аргументной и критериальной оптимизации, оптимальные на классе функции потерь оказываются различными. Поэтому приведем две формулировки теоремы об оптимальных на классе функциях потерь.

Теорема 4 (об аргументно оптимальной на классе функции потерь и ее производной).

Аргументно оптимальные на классе функции F_* и их производные F_*' равны

$$F_*[\varepsilon] = -\ln p_*(\xi) |_{\xi=\varepsilon}, \quad (107)$$

$$F_*'[\varepsilon] = -\frac{p_*'(\xi)}{p_*(\xi)} |_{\xi=\varepsilon}, \quad (108)$$

где $p_*(\xi)$ — наименее благоприятная плотность распределения

$$p_*(\xi) = \arg \min_{p_0 \in P} I(p_0) \operatorname{tr} A(\Theta^*, \sigma^2(p_0)), \quad (109)$$

удовлетворяющая принципу аргументной оптимизации на классе

$$\text{tr } V^{-1}(F_*', p_0) \geq \text{tr } V^{-1}(F_*, p_*) = \text{tr } V^{-1}(p_*) \geq \text{tr } V^{-1}(F', p_*) \quad (110)$$

при всех $p_0(\xi) \in P$ и $F = -\ln p$.

Доказательство основано на минимизации $\text{tr } V^{-1}(F_*', p_0)$ и $\text{tr } V^{-1}(p_0)$ по p_0 и установлении того, что они достигают одного и того же минимума при $p_0(\xi) = p_*(\xi)$. Отсюда следует правая часть неравенства (110). Из условий оптимальности (78) следует правая часть неравенства (110). А (109) следует из минимизации $\text{tr } V^{-1}(p_0)$ по $p \in P$.

Принцип оптимальности на классе означает, что оптимальные на классе функции потерь $F_*[\varepsilon]$ и их производные $F_*'[\varepsilon]$ гарантируют, что при любой плотности распределения помех АМК $V(F_*', p_0)$ и АУ $w(F_*', p_0)$ не превзойдут $V(F_*', p_*)$ и $w(F_*', p_*)$, которые являются минимальными среди всех $V(F', p_*)$ и $w(F', p_*)$. Заметим, что в общем случае аналитически определить $p_*(\xi)$ (105) не представляется возможным и поэтому приходится прибегать к разного рода численным методам. Только для класса P_2 наименее благоприятная плотность распределения (109), которая с учетом (57) представится в виде

$$p_*(\xi) = \arg \min_{p_0 \in P_2} I_F(p_0) [\text{tr } A_I(\Theta^*) + \sigma^2(p_0) \text{tr } A_{II}(\Theta^*)],$$

представляет собой нормальную плотность распределения $N(0, \sigma_1^2)$, которая одновременно минимизирует $I_F(p_0)$ и $I_F(p_0)\sigma^2(p_0)$. Таким образом, для класса P_2 аргументно оптимальная на классе функция потерь квадратична, а ее производная линейна, т. е.

$$F_*[\varepsilon] = \frac{\varepsilon^2}{2\sigma_1^2} + \ln 2\pi\sigma_1^2, \quad F_*'[\varepsilon] = \frac{\varepsilon}{\sigma_1^2}.$$

Теорема 5 (о критериально оптимальной на классе функции потерь и ее производной).

Критериально оптимальные на классе функция потерь и ее производная равны

$$F_*[\varepsilon] = -\ln p_*(\xi) \Big|_{\xi=\varepsilon} \quad (111)$$

и

$$F_*'[\varepsilon] = -\frac{p_*'(\xi)}{p_*(\xi)} \Big|_{\xi=\varepsilon} \quad (112)$$

где

$$p_*(\xi) = \arg \min_{p_0 \in P} I_F(p_0), \quad (113)$$

удовлетворяющая принципу критериальной оптимизации

$$w(F_*, p_0) \leq w(F_*, p_*) = w(p_*) \leq w(F_*, p_0) \quad (114)$$

для всех $p_0 \in P$ и $F = -\ln p$.

Доказательство здесь аналогично предыдущему

Определение $p_*(\xi)$ (113) теперь несоизмеримо проще, и во многих случаях существуют аналитические решения [17, 60—63]. Примеры критериально оптимальных на классе функций потерь $F_*[\varepsilon]$, их производных $F_*'[\varepsilon]$ и соответствующих наименее благоприятных плотностей распределения $p_*(\xi)$ приведены в табл. 2.

Для получения абсолютно оптимальных на классе алгоритмов идентификации достаточно в абсолютно оптимальных ал-

Таблица 2

№	CLASS \mathcal{P}	$p_*(\xi) = \arg \min_{p \in \mathcal{P}} I_F(p)$	$F_*[\varepsilon] = -\ln p_*(\xi) \Big _{\xi=\varepsilon}$	$F_*'[\varepsilon] = -\frac{p_*'(\xi)}{p_*(\xi)} \Big _{\xi=\varepsilon}$
1	$\mathcal{P}_1: p_0(0) \geq \frac{1}{2\sigma_1} > 0$			
2	$\mathcal{P}_2: \sigma^2(p_0) \leq \sigma_1^2$			
3	$\mathcal{P}_3: p_0(\xi) \geq (1-\alpha)N(0, \sigma_N^2)$			
4	$\mathcal{P}_4: p_0(\xi) \geq (1-\alpha)R(0, \rho_N)$			

горитмах (94) заменить $F_0'[\varepsilon]$ и $F_0''[\varepsilon]$ на $F_*'[\varepsilon]$ и $F_*''[\varepsilon]$, соответствующих априорно заданному классу. Таким образом, абсолютно оптимальные на классе алгоритмы принимают вид

$$\begin{aligned} \Theta_n &= \Theta_{n-1} + \hat{\Gamma}_{0,n} F_*'[\varepsilon_n^c(\Theta_{n-1})] v_n, \\ \hat{\Gamma}_{0,n}^{-1} &= \hat{\Gamma}_{0,n-1}^{-1} + F_*''[\varepsilon_n^c(\Theta_{n-1})] v_n v_n^T, \\ v_n &= (1 - \hat{S}_{(k)}(q)) v_n + x_n^{(k)}. \end{aligned} \quad (115)$$

Конкретный вид прямых ($k=0$) и непрямых алгоритмов ($k=1, 2$) получается из (115) при замене $\hat{S}_{(k)}(q)$ его значением из (44) и дополнением их выражением невязки. Они также получаются из (96)–(98) после замены $F_0'[\varepsilon]$ и $F_0''[\varepsilon]$ на $F_*'[\varepsilon]$ и $F_*''[\varepsilon]$.

Алгоритмы (115) наиболее общие. Если класс P сводится к единственной плотности распределения, то из них вытекают абсолютно оптимальные алгоритмы. Для класса P_2 -плотностей распределения с ограниченной дисперсией, как видно из табл. 2, мы приходим к квадратичному критерию и линейным по невязке алгоритмам. Для иных классов критерии отличны от квадратичных, и алгоритмы уже нелинейны по невязке.

Оптимальность на классе, робастность, регуляризация. Рассмотрим статистическую интерпретацию оптимальных и оптимальных на классе функций потерь. Как следует из определения оптимальной функции потерь (73), она представляет собой не что иное, как функцию правдоподобия [7, 57] с обратным знаком. Поэтому отыскание оценок оптимального решения соответствует методу максимального правдоподобия (ММП), который широко используется при решении разнообразных задач идентификации [4, 5, 8, 9, 37, 53]. Установленная связь показывает оптимальность ММП в смысле асимптотической эффективности и позволяет на ММП взглянуть с иной точки зрения.

Для квадратичной функции потерь, соответствующей нормальной плотности распределения помех, ММП приводит по существу к методу наименьших квадратов (МНК). К использованию МНК по существу и призывают почти все авторы цитат, приведенных в п. 3, и многочисленные их последователи. Но как уже давно выяснили статистики, при «малом» отклонении плотности распределения помех $p_0(\xi)$ от нормальной МНК может стать неработоспособным. Это свидетельствует о сильной чувствительности МНК к упомянутому отклонению, т. е. о его нестабильности или негрубости.

Большое значение в выяснении негрубости МНК в задаче оценивания параметра сдвига и в пропаганде необходимости новых подходов сыграли работы Тьюки [64, 65]. Основоположником такого нового подхода, получившего название робастного оценивания, является Хубер [63]. Он не только предложил

новые оценки параметра сдвига, обладающие грубостью, но и обосновал их оптимальность в минимаксном смысле. Робастному оцениванию, его состоянию и развитию посвящены работы [66–69]. Наиболее полный обзор огромного числа работ по робастному оцениванию и его обобщениям приведен в [70]. Отметим также сборники и монографии [71–77], посвященные этому кругу вопросов.

Нестабильность, негрубость МНК, а в ряде случаев и ММП, вызвана сильным влиянием на оценку Θ_n малых отклонений используемой априорной информации о помехах от действительной. Так, для выбора оптимальной функции потерь $F_0[\varepsilon]$ необходимо точно знать плотность распределения помех $p_0(\xi)$. Малые отклонения от этой информации могут не только сделать оценки далекими от оптимальной, но и вообще бессмысленными. Покажем это на примере. Пусть предполагаемая плотность распределения, как это часто принимается, нормальная

$$p_a(\xi) = N(0, \sigma_N^2), \quad (116)$$

а истинная плотность распределения представляет собой так называемую загрязненную нормальную плотность распределения

$$p_0(\xi) = (1-\alpha)p_a(\xi) + \alpha p_1(\xi), \quad (117)$$

где $p_1(\xi)$ — произвольная плотность распределения помех с дисперсией $\sigma_1^2 \gg \sigma_N^2$, $\alpha \ll 1$, $\alpha \sigma_1^2 \gg \sigma_N^2$. Заметим, что $p_0(\xi)$ принадлежит классу P_2 . По предполагаемой плотности распределения $p_a(\xi)$ (116) определяется оптимальная функция потерь и производная

$$F_a[\varepsilon] = -\ln p_a(\xi)|_{\xi=\varepsilon}, \quad F_a'[\varepsilon] = -\frac{p_a'(\xi)}{p_a(\xi)} \Big|_{\xi=\varepsilon} \quad (118)$$

Рассмотрим теперь соотношения между АМКО $V(F_a', p_0)$ и $V(p_a)$.

И удивимся, что $V(F_a', p_0)$ частично много больше, чем

$$V(F_a', p_0) \gg V(p_a), \quad (119)$$

и некоторого числа $\beta \gg 1$ имеет место матричное нера-

$$V(F_a', p_0) \gg \begin{vmatrix} I & 0 & 0 \\ 0 & \beta I & 0 \\ 0 & 0 & I \end{vmatrix} V(p_a), \quad (120)$$

квивалентно,

$$V^{-1}(F_a', p_0) \gg \begin{vmatrix} I & 0 & 0 \\ 0 & \beta^{-1} I & 0 \\ 0 & 0 & I \end{vmatrix} V^{-1}(p_a). \quad (121)$$

т. е. матрица размерности N .

Из выражений для АМКО (68), (79) и структуры НИМ (57) нетрудно сделать заключение, что при

$$\beta = \frac{\sigma^2(p_0)}{\sigma^2(p_2)} \gg 1 \quad (122)$$

неравенства (120), (121) выполняются. А это значит, что малое отклонение предполагаемой плотности распределения $p_a(\xi)$ от истинной приводит к существенному частичному увеличению АМКО, а, значит, к существенному частичному уменьшению скорости сходимости соответствующих оценок, а именно, оценок вектора параметров b^* . Отсюда следует, что МНК груб при оценивании векторов параметров a^* , c^* и не груб при оценивании вектора параметров b^* . Таким образом, для квадратичной функции потерь $F_N[\varepsilon] = -\ln p_N(\xi)_{\xi=\varepsilon}$, которую мы считаем оптимальной, согласованной, часть получаемых оценок не грубы, нестабильны, а в ряде случаев несостоятельны.

Причиной негрубости, нестабильности оценок МНК является неадекватность используемой квадратичной функции потерь реальной ситуации, а даже малые отклонения $p_0(\xi)$ от предполагаемой плотности распределения $p_a(\xi)$ могут привести к резкому ухудшению точности оценок. Поэтому более реалистичский подход состоит в использовании имеющейся неполной априорной информации относительно плотности распределения, которая задается классом, т. е. использовании оптимизации на классе.

Использование оптимальной на классе функции потерь $F_*[\varepsilon] = -\ln p_*(\xi)_{\xi=\varepsilon}$ устраняет негрубость. Как следует из принципа оптимальности на классе, АМКО $V(F_*', p_0)$ ограничена сверху АМКО $V(F_*', p_*) = V(p_*)$. Подобная ситуация, несмотря на различие конкретных оптимальных на классе функций потерь, остается справедливой и при критериальной оптимизации. Критериально оптимальные на классе функции потерь ограничивают АУ $\omega(F', p_0)$ величиной $V(p_*)$. Такие оценки и методы можно назвать робастными [62—77].

До сих пор всегда предполагалось, что НИМ — положительно определенная матрица. Это предположение выполняется далеко не всегда. Дело в том, что НИМ зависит от наблюдений, которыми не всегда можно управлять при идентификации в режиме нормальной работы. Поэтому часто можно лишь предполагать, что НИМ является неотрицательно определенной, т. е.

$$A(\Theta^*, \sigma^2(p_0)) = E\{v_n v_n^T\} \geq 0. \quad (123)$$

Это значит, что НИМ вырождена или близка к вырожденной. Обращение такой матрицы, фигурирующее в алгоритмах, приводит к их неработоспособности. Одна из возможностей борьбы с обращением плохо обусловленной матрицы состоит в

использовании метода регуляризации [78, 79]. Идея метода регуляризации состоит в улучшении обусловленности НИМ путем добавления к ней положительно определенной матрицы, которая с течением времени стремится к нулю. Нетрудно видеть, что такого рода регуляризацию осуществляют положительно определенные начальные условия $\Gamma_0 > 0$ рекуррентного соотношения матрицы усиления. К регуляризации следует прибегать лишь тогда, когда по существу задачи идентификации необходимо определить как можно точнее оценки параметров динамического объекта.

Если же нам надо построить и использовать модель динамического объекта, то целесообразно от аргументной оптимизации перейти к критериальной оптимизации, в которой вырожденность НИМ не играет роли. В вырожденном случае рекуррентное уравнение относительно матрицы усиления в критериально оптимальных на классе алгоритмах осуществляет рекуррентно операцию псевдообращения матрицы. При этом АУ, определяемое (65) вместо (80) становится равным

$$\omega(p_0) = \frac{\tilde{N}}{I_F(p)}, \quad \tilde{N} = \text{rank } A(\Theta^*, \sigma^2(p_0)) < N_c. \quad (124)$$

Следовательно, АУ для вырожденного случая уменьшается. Физически это очевидно, т. к. плохая обусловленность вызывается малостью невязки. Применение критериально оптимальных алгоритмов на классе в этом случае гарантирует минимизацию критерия, и чем больше вырожденность, тем меньше АУ.

Сказанное выше, а также относительная простота критериальных алгоритмов позволяют рекомендовать их для использования не только в вырожденных, но и в невырожденных случаях.

Упрощенные алгоритмы. Рассмотрим условие оптимальности (48) при $F[\varepsilon] = F_*[\varepsilon]$

$$VJ(\Theta) = -E\{F_*'[\varepsilon_n(\Theta)]v_n\} = 0, \quad (125)$$

где функция чувствительности равна (47)

$$v_n = (1 - \hat{S}_{(k)}(q))v_n + x_n^{(k)}. \quad (126)$$

Подставляя v_n из (126) в (125), получим

$$\begin{aligned} \nabla J = & -\bar{E}\{F_*'[\varepsilon_n^C(\Theta)](1 - \hat{S}_{(k)}(q))\}, \\ & -\bar{E}\{F_*'[\varepsilon_n^C(\Theta)]x_n^{(k)}\} = 0. \end{aligned} \quad (127)$$

При $\Theta = \Theta^*$ невязка $\varepsilon_n(\Theta^*) = \xi_n$ (23). Поэтому

$$\begin{aligned} \nabla J = & -\bar{E}\{F_*'[\xi_n](1 - \hat{S}_{(k)}(q))\}, \\ & -\bar{E}\{F_*'[\xi_n]x_n^{(k)}\} = 0. \end{aligned} \quad (128)$$

В силу независимости помехи ξ_n и наблюдений $x_n^{(k)}$, а значит и v_n , из (128) следуют равенства

$$-\bar{E}\{F_*'[\xi_n]x_n^{(k)}\}=0, \quad (129)$$

$$-\bar{E}\{F_*'[\xi_n](1-\hat{S}_{(k)}(q))v_n\}=0. \quad (130)$$

Фактически эти равенства ранее (см. п. 5) были использованы для установления факта инвариантности оптимального решения Θ^* относительно симметричных функций потерь. Здесь мы используем (129) для построения упрощенных алгоритмов.

Как следует из (129), условие оптимальности (125) можно заменить более простым.

$$-E\{F_*'[\varepsilon_n^c(\Theta)]x_n^{(k)}\}=0. \quad (131)$$

Левую часть (131) можно рассматривать как псевдоградиент [9]. Условие (131) порождает алгоритмы, которые отличаются от оптимальных на классе алгоритмов тем, что вместо функции чувствительности теперь в них фигурируют непосредственно наблюдения.

Псевдоградиентные алгоритмы представляются в виде

$$\begin{aligned} \Theta_n &= \Theta_{n-1} + \hat{\Gamma}_{0,n} F_*' [\varepsilon_n^c(\Theta_{n-1})] x_n^{(k)}, \\ \hat{\Gamma}_{0,n}^{-1} &= \hat{\Gamma}_{0,n-1}^{-1} + F_*'' [\varepsilon_n^c(\Theta_{n-1})] x_n^{(k)} x_n^{(k)T}. \end{aligned} \quad (132)$$

Приведем конкретный вид этих алгоритмов, пользуясь обозначением невязок $\varepsilon_n(\Theta)$ и $\hat{S}_{(k)}(q)$. Прямые алгоритмы

$$\begin{aligned} \Theta_n^{(0)} &= \Theta_{n-1}^{(0)} + \hat{\Gamma}_{0,n} F_*' [\varepsilon_n^c(\Theta_{n-1}^{(0)})] x_n^{(0)}, \\ \hat{\Gamma}_{0,n}^{-1} &= \hat{\Gamma}_{0,n-1}^{-1} + F_*'' [\varepsilon_n^c(\Theta_{n-1}^{(0)})] x_n^{(0)} x_n^{(0)T}, \\ \varepsilon_n^c(\Theta_{n-1}^{(0)}) &= y_n - (C^0(q) - 1)y_n + \Theta_{n-1}^{(0)T} x_n^{(0)}. \end{aligned} \quad (133)$$

Непрямые алгоритмы первого рода

$$\begin{aligned} \Theta_n^{(1)} &= \Theta_{n-1}^{(1)} + \hat{\Gamma}_{0,n} F_*' [\varepsilon_n^c(\Theta_{n-1}^{(1)})] x_n^{(1)}, \\ \hat{\Gamma}_{0,n}^{-1} &= \hat{\Gamma}_{0,n-1}^{-1} + F_*'' [\varepsilon_n^c(\Theta_{n-1}^{(1)})] x_n^{(1)} x_n^{(1)T}, \\ \varepsilon_n^c(\Theta_{n-1}^{(1)}) &= y_n - \Theta_{n-1}^{(1)T} x_n^{(1)}. \end{aligned} \quad (134)$$

Непрямые алгоритмы второго рода

$$\begin{aligned} \Theta_n^{(2)} &= \Theta_{n-1}^{(2)} + \hat{\Gamma}_{0,n} F_*' [\varepsilon_n^c(\Theta_{n-1}^{(2)})] x_n^{(2)}, \\ \hat{\Gamma}_{0,n}^{-1} &= \hat{\Gamma}_{0,n-1}^{-1} + F_*'' [\varepsilon_n^c(\Theta_{n-1}^{(2)})] x_n^{(2)} x_n^{(2)T}, \\ \varepsilon_n^c(\Theta_{n-1}^{(2)}) &= y_n - \Theta_{n-1}^{(2)T} x_n^{(2)}. \end{aligned} \quad (135)$$

Прямые алгоритмы (132) при $C^0(q) = 1$, $F_*'[\varepsilon] = \varepsilon$, $F_*''[\varepsilon] = 1$ совпадают с линейными алгоритмами ММП метода предсказания ошибки псевдолинейной регрессии [9], которые, таким образом, являются оптимальными на классе P_2

При ином выборе матриц усиления мы приходим к различным иным алгоритмам, например типа алгоритмов стохастической аппроксимации [80, 81]

Приведем наиболее употребительные правила формирования последовательности «матриц шагов» $\{\Gamma_n\}$, которые порождают разные упрощенные алгоритмы:

а) алгоритм стохастической аппроксимации [82]

$$\Gamma_n = \gamma_n I, \quad \gamma_n = \gamma_0 n^{-1}, \quad \gamma_0 > 0 \quad (136)$$

б) алгоритм Гудвина [83]

$$\Gamma_n = \gamma_n I, \quad \gamma_n = \gamma_0 \left(1 + \sum_{i=1}^n \|x_i^{(0)}\|^2 \right)^{-1} \quad (137)$$

в) алгоритм расширенного МНК [84]

$$\begin{aligned} \Gamma_n &= \left(\rho_0 I + \sum_{i=1}^n x_i^{(0)} x_i^{(0)T} \right)^{-1} = \Gamma_{n-1} - \frac{\Gamma_{n-1} x_n^{(0)} x_n^{(0)T} \Gamma_{n-1}}{1 + x_n^{(0)T} \Gamma_{n-1} x_n^{(0)}}, \\ \Gamma_0 &= \rho_0^{-1} I, \quad \rho_0 > 0. \end{aligned} \quad (138)$$

Алгоритмы типа а), б) относятся к так называемым *SG*-алгоритмам, т. е. к алгоритмам со «скалярным шагом», или к алгоритмам типа «стохастического (S) градиента (G)». Они характеризуются простой программной реализацией. Алгоритмы типа в) требуют пересчета всей матрицы шагов Γ_n и поэтому обычно более трудоемки в смысле программной или аппаратной реализации. Разумеется, они уже отличаются от абсолютно оптимальных или абсолютно оптимальных на классе тем, что их скорость сходимости ниже. Потерю скорости сходимости можно подсчитать, хотя это не всегда легко сделать, пользуясь уравнениями относительно АМКО и АУ в общем случае.

Анализ свойств упрощенных алгоритмов. При анализе свойств упрощенных рекуррентных алгоритмов потребуется несколько определений, которые и приводятся ниже.

Определение 1. Устойчивый оператор $\tilde{P}_{\xi}(q) = \frac{P_{\xi,1}(q)}{P_{\xi,2}(q)}$ будем называть *SRP*-оператором (strictly real positive [85]), или пассивным оператором, если он имеет строго положительную вещественную часть, т. е. для всех $\omega \in [0, 2\pi]$

$$2 \operatorname{Re} \tilde{P}_{\xi}(e^{i\omega}) = \tilde{P}_{\xi}(e^{i\omega}) + \tilde{P}_{\xi}(e^{-i\omega}) > 0. \quad (139)$$

Отметим, что если $\tilde{P}_{\xi}(q) = \tilde{P}_{\xi,1}(q)$ есть *SRP*-оператор, то из этого следует, что многочлен $P_{\xi,1}(q)$ устойчив. Обратное неверно. На рис. 1а представлено множество параметров, характеризующих *SRP*-операторы $\tilde{P}_{\xi}(q) = P_{\xi,1}(q) = 1 + c_1^* q + c_2^* q^2$, а на рис. 1б — те же *SRP*-операторы, но в форме $\tilde{P}_{\xi}(q) = P_{\xi,1}(q) = (1 + a_1^*)(1 + a_2^*)$.

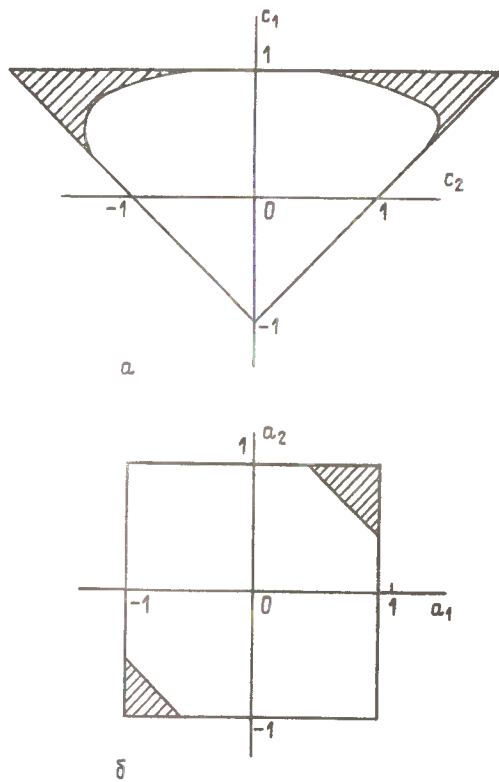


Рис. 1

Определение 2. Нелинейное преобразование $\psi(x): R^1 \rightarrow R^1$ будем называть квазилинейным относительно SRP -оператора $\tilde{P}_{\xi}(q) = P_{\xi,1}(q)$, если при всех $n = 1, 2, \dots$

$$\sum_{i=0}^n \psi(x_i) y_i \geq \mu_0 \sum_{i=0}^n x_i^0 - C, \quad (140)$$

где $\{y_i\}, \{x_i\}$ связаны соотношением

$$y_n = \tilde{P}_{\xi}(q) x_n, \quad (141)$$

причем

$$\mu_0 = 0, \quad (142)$$

а $C \geq 0$ — линейная функция от нормы вектора начальных условий

$$\left(C = \sum_{i=1}^{N_2} \psi_i(x_i) \sum_{s=1}^{N_2} |c_s| |x_{i-s}| \right).$$

Приведем далее утверждение, позволяющее конструктивно выделять преобразования $\psi(x)$, заведомо содержащиеся в классе квазилинейных.

Теорема 6 (достаточные условия квазилинейности):

Для того, чтобы преобразование $\psi = \psi(x)$ было квазилинейным относительно SRP -оператора $P_{\xi,1} = I + \sum_{i=1}^{N_2} c_i^* q^i$, достаточно потребовать выполнения следующего условия «секторности»:

$$\alpha x^2 \leq x\psi(x) \leq \beta x^2, \quad (143)$$

где α, β — неотрицательные числа, удовлетворяющие условию

$$1 \geq \frac{\alpha}{\beta} > \frac{\lambda_1 - \lambda_0}{\lambda_1 + \lambda_0}, \quad (144)$$

где

$$0 < \lambda_0 \triangleq \min_{\omega} \operatorname{Re} P_{\xi,1}(e^{i\omega}) \leq \lambda_1 \triangleq 1 + \sum_{i=1}^{N_2} |c_i^*|. \quad (145)$$

Проиллюстрируем теперь зависимость раствора сектора (143) допустимых нелинейностей от коэффициентов c_i^* полинома $P_{\xi,1}(q)$ при $N_2 = 1$.

В этом случае имеем:

$$P_{\xi,1}(q) = 1 + c_1^* q, \quad 0 < \lambda_0 = 1 - |c_1^*|, \quad \lambda_1 = 1 + |c_1^*|.$$

Тогда требование (144) утверждения сводится к выбору α и β из условия

$$1 \geq \frac{\alpha}{\beta} > |c_1^*|,$$

т. е. при заданных α и d_1 допустимы все β такие, что

$$\alpha \leq \beta < \frac{\alpha}{|c_1^*|}.$$

Таким образом, при уменьшении зависимости (коррелированности) действующей помехи

$$\xi_n = P_{\xi,1}(q) \xi_n = \xi_n + c_1^* \xi_{n-1},$$

т. е. при $|c_1^*| \rightarrow 0$ расширяется диапазон допустимых нелинейностей. При $|c_1^*| \rightarrow 1$ допустимый диапазон для β резко сужается, приближаясь к линейной зависимости. Чем больше запас устойчивости фильтра $P_{\xi,1}(q)$, формирующего помеху, тем более широкий диапазон нелинейностей $\psi(x)$ допустим соответствующим условием «секторности» (143).

Используя понятие «секторности» и теорему 6, сформулируем основную результат о сходимости беспроекционных алгоритмов типа «стохастического градиента» со скалярным (возможно, зависимым от имеющихся наблюдений) шагом γ_n . Эти

алгоритмы имеют вид

$$\begin{aligned}\Theta_n &= \Theta_{n-1} + \gamma_n x_n^{(0)} \varphi(\varepsilon_n), \\ \varepsilon_n &= y_n - \Theta_{n-1}^T x_n^{(0)}.\end{aligned}\quad (146)$$

Теорема 7 (о сильной состоятельности SG-алгоритмов).

Пусть для алгоритмов (146) выполнены следующие условия:

- 1) известны степени N, N_1, N_2 соответствующих полиномов в (146);
- 2) существует константа $k_0 < 0$, такая, что для всех $t = 1, 2, \dots$

$$\gamma_t x_t^{(0)T} x_t^{(0)} \stackrel{\text{п.н.}}{\leq} k_0 < \infty;$$

$$3) \sum_{t=1}^{\infty} \gamma_t^2 x_t^{(0)T} x_t^{(0)} > \infty;$$

$$4) \gamma_t \stackrel{\text{п.н.}}{\leq} \gamma_t \text{ для всех } t = 1, 2, \dots;$$

5) $\{\xi_t\}$ — последовательность независимых, одинаково распределенных центрированных случайных величин с ограниченным вторым моментом;

6) функция $\varphi(x)$ такова, что существуют

$$E\{\varphi(x + \xi_t)\} = \psi(x),$$

$$E\{\varphi^2(x + \xi_t)\} \leq k_1 + k_2 x \psi(x), \quad k_1, k_2 < \infty$$

для всех $x \in R^1$;

7) функция $\psi(x)$ задает «секторное» преобразование относительно динамического SRP-оператора $[P_{\xi,1}(q) - 1/2k_0k^2]$.

Тогда справедливы следующие свойства:

$$\text{I) } \|\Theta_n\| \stackrel{\text{п.н.}}{\leq} C(\omega) < \infty;$$

$$\text{II) } \|\Theta_n - \Theta_{n-1}\| \stackrel{\text{п.н.}}{\rightarrow} 0;$$

$$\text{III) } \sum_{t=1}^{\infty} \gamma_t (\varepsilon_t - \xi_t)^2 \stackrel{\text{п.н.}}{<} \infty;$$

$$\text{IV) } \|\Theta_n - \Theta\|^2 \stackrel{\text{п.н.}}{\rightarrow} V_0(\omega) < \infty.$$

Если к тому же дополнительно выполняется требование «достаточного разнообразия входов»

$$8) \sum_{l=0}^{\infty} \lambda_{\min} \left(\sum_{j=1}^r \gamma_{l+j} x_{l+j}^{(0)} x_{l+j}^{(0)T} \right) \stackrel{\text{п.н.}}{=} \infty, \quad r > N + N_1 + N_2,$$

то

$$\text{V) } \Theta_n \stackrel{\text{п.н.}}{\rightarrow} \Theta.$$

Рассмотрим подробнее, во что переходят условия приведенного выше утверждения для двух наиболее часто встречающихся типов SG-алгоритмов.

Следствие 1. Пусть в алгоритме (146) последовательность $\{\gamma_n\}$ выбрана в виде (136), а именно

$$\Gamma_n = \gamma_n I, \quad \gamma_n = \gamma_0 n^{-1}, \quad \gamma_0 > 0.$$

Если при выполнении условий 1), 5), 6) теоремы 7 дополнительно

а) полином $Q(q)$ системы (146) устойчив;

б) выполнено условие «достаточного разнообразия входов», то есть

$$E\{\xi_n^2\} = \sigma_{\xi}^2 > 0, \quad \lim_{i \rightarrow \infty} \lambda_{\min} \left(\sum_{j=1}^r x_{i+r+j} x_{i+r+j}^T \right) \stackrel{\text{п.н.}}{>} 0;$$

в) $\psi = \psi(x)$ задает «секторное» преобразование относительно SRP-оператора $P_{\xi,1}(q)$,

то для почти всех $\omega \in \Omega_0$ имеем $\Theta_n \stackrel{\text{п.н.}}{\rightarrow} \Theta$, где

$$\Omega_0 = \{\omega \in \Omega \mid \overline{\lim}_{n \rightarrow \infty} (|u_n| + |\xi_n|) \leq \text{const} < \infty\}.$$

Связь условий этого следствия с условиями 2), 3), 4), 7), 8) теоремы достаточно очевидна. Так, из (136) и а) следует условие 2) утверждения (при бесконечно малом k_0) и условия 3), б) и в) при малых k_0 вытекает требование 7). В свою очередь, из (136) и условия б) следует из (136) и условия б).

Таким образом, гарантировать сходимость алгоритма стохастической аппроксимации (146) в этих задачах можно лишь на множестве реализаций, соответствующих ограниченному по величине возмущениям $\{\xi_n\}$ и управляющим воздействиям $\{u_n\}$. (Следствие 2 (о сходимости алгоритма Гудвина). Пусть в алгоритме (146) последовательность $\{\gamma_n\}$ выбрана в виде 7) и именно

$$\gamma_n = \gamma_0 \left(1 + \sum_{t=1}^n \|x_t^{(0)}\|^2 \right)^{-1}, \quad \gamma_0 > 0.$$

и при выполнении 1), 5) 6) теоремы 7 дополнительно

а) выполнено условие «достаточного разнообразия входов»,

$$E\{\xi_n^2\} = \sigma_{\xi}^2 > 0, \quad \lim_{i \rightarrow \infty} \lambda_{\min} \left(\sum_{j=1}^r x_{i+r+j} x_{i+r+j}^T \right) \stackrel{\text{п.н.}}{>} 0;$$

то преобразование $\psi = \psi(x)$ является «секторным» относительно оператора

$$P_{\xi,1}(q) \text{ при устойчивом полиноме } Q(q)$$

$[P_{\varepsilon,1}(q) - 1/2\gamma_0 K_2]$ при любом (в том числе неустойчивом) полиноме $Q(q)$,

то для почти всех $\omega \in \Omega_0 \subseteq \Omega$ имеем

$$\Theta_n \xrightarrow[n \rightarrow \infty]{\text{п.н.}} \Theta,$$

где

$$\Omega_0 = \begin{cases} \omega \in \Omega \mid \overline{\lim}_{n \rightarrow \infty} (|u_n| + |\xi_n|) \leq \text{const} \text{ при устойчивых } Q(q) \\ \Omega \text{ при любых } Q(q) \text{ и SRP-операторах } [P_{\varepsilon,1}(q) - \frac{1}{2}\gamma_0 k_2]. \end{cases}$$

Заметим, что условие а) гарантирует, что $\sum_{i=0}^n \|x_i^{(0)}\|_{n \rightarrow \infty}^2 \xrightarrow{\text{п.н.}} \infty$,

откуда в силу леммы Абеля—Дини [86] вытекает требование 3) теоремы. Если автономный объект устойчив, т. е. устойчив полином $Q(q)$, то условие 2) выполняется автоматически при бесконечно малых k_0 , начиная с некоторого конечного $n_0(\omega)$. В общем же случае, когда автономный объект не обязательно устойчив (но стабилизируем за счет выбора управлений $\{u_n\}$), справедлива оценка $\gamma_0 \leq k_0$ и при более жестких требованиях на «цветность» помехи ξ_n а именно при SRP-условии на полином $[P_{\varepsilon,1}(q) - \frac{1}{2}\gamma_0 k_2]$ на множестве ограниченных помех $\{\xi_n\}$ и управлений $\{u_n\}$ гарантируется выполнение условия 2). Условие б) обеспечивает выполнение требования 8) теоремы.

Таким образом, для любого полинома $P_{\varepsilon,1}(q)$, обладающего ε -запасом по устойчивости ($\varepsilon > 0$) можно подобрать достаточно малое $\gamma_0 > 0$, при котором алгоритм Гудвина (137), (146) обеспечивает гарантированное (с вероятностью 1) оценивание параметров объекта (включая параметры $\{c_i^*\}$ цветности помехи). Естественно, помехи $\{\xi_n\}$ и возмущения $\{u_n\}$ при этом должны обеспечивать условие достаточного разнообразия входов.

Как показано в [82], решение вариационной задачи (109) часто реализуется на классе функций $\varphi^*(v)$ типа знаковых, релейных и типа «упор». Однако такие функции не удовлетворяют условию (144), в котором требуется принадлежность осредненной нелинейности $\psi(v)$ существенно положительному сектору (143) с коэффициентом $\alpha > 0$. Таким образом, оказалось, что нелинейности, рост которых происходит медленнее некоторого линейного закона, никогда не принадлежат никакому положительному (α, β) -сектору, а следовательно, для них нельзя гарантировать сходимость соответствующих процедур. Однако можно выделить более узкий, чем SRP-класс операторов $P_{\varepsilon,1}(q)$, гарантирующих сильную состоятельность и для секторов с $\alpha = 0$

Теорема 8 (о релейных нелинейностях) Пусть функция $\psi = \psi(x)$, $x \in R^1$ такова, что

$$0 \leq x\psi(x) \leq \beta x^2 \quad (\beta \in (0, \infty)),$$

$$\psi(-x) = -\psi(x),$$

$\psi(x)$ — дифференцируема всюду в R^1 , $\psi'(x) > 0$ для всех $x \in R^1$ и дважды дифференцируема в точке $x = 0$. Тогда для выполнения условия (140) квазилинейности этой функции относительно оператора $P_{\varepsilon,1}(q)$ достаточно, чтобы

$$\sum_{i=1}^{N_2} |c_i^*| < 1.$$

Следствие. При выполнении условий теорем 7, 8 SG-алгоритмы с вероятностью 1 гарантируют сходимость оценок $\Theta_n \rightarrow \Theta$.

Рассмотрим в качестве примера SG-алгоритм, содержащий знаковую нелинейность

$$\varphi(\varepsilon) = \text{sign } \varepsilon.$$

Функция $\psi(x)$, соответствующая этой нелинейности, записывается в форме

$$\psi(x) \stackrel{\Delta}{=} E\{\varphi(x + \xi_n)\}$$

и имеет вид (при гауссовых ξ_n), изображенный на рис. 2. Если ξ_n распределены, например по Лапласу $p(\xi) = \frac{1}{2} \exp(-|\xi|)$, то $\psi(x) = (1 - \exp(-|x|)) \text{sign } x$. Как видно из приведенного

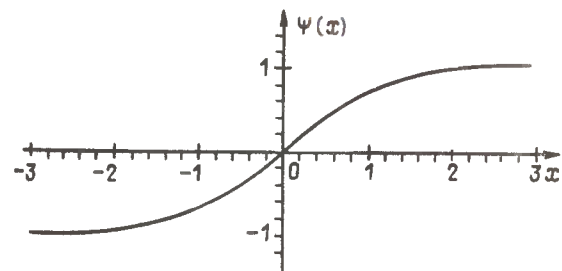


Рис. 2

рисунка, не существует $\alpha > 0$ такого, что $\alpha x^2 \leq x\psi(x)$ при всех $x \in R^1$. Но тем не менее, выполнены условия теоремы 8, в силу которой работоспособность этого алгоритма с такой нелинейностью, например при $N_2 = 2$, гарантируется при всех c_i^* ($i = 1, 2$), лежащих внутри ромба (см. рис. 3). Множество D коэффициентов c_1, c_2 , удовлетворяющих условиям теоремы 8, лежит внутри множества D_{SRP} коэффициентов, соответствующих SRP-многочленам $P_{\varepsilon,1}(q)$, которое, в свою очередь, располагает

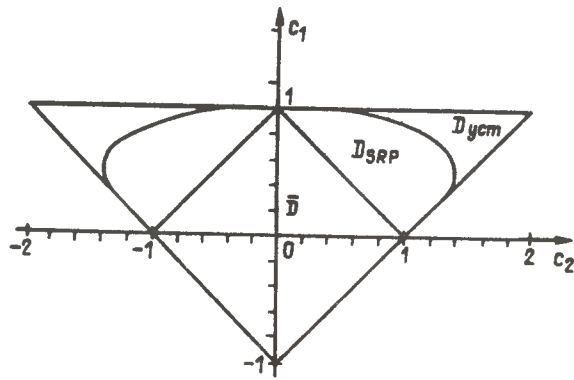


Рис. 3

ся внутри множества $D_{уст}$ коэффициентов устойчивых многочленов $P_{\xi,1}(q)$. При $N_2=1$ все эти множества совпадают, а при $N_2 \geq 2$

$$\bar{D} \subseteq D_{SRP} \subseteq D_{уст}.$$

Распространим теперь аналогичный подход на проекционную версию алгоритма (133) с матричным коэффициентом Γ_n , который формируется по правилу (138), где оператор π_n осуществляет проектирование на ограниченное множество C^* векторов Θ , последние N_2 компонент которых соответствуют множеству устойчивых с ε -запасом многочленов $P_{\xi,1}(q)$

$$\Theta_n = \pi_n \{ \Theta_{n-1} + \hat{\Gamma}_{0,n} F_*' [\varepsilon_n^C(\Theta_{n-1})] x_n^{(0)} \}$$

по норме, задаваемой матрицей

$$S_n = (r_n \Gamma_n)^{-1}, \quad r_n \triangleq \text{tr} \Gamma_n^{-1} = (N + N_1 + N_2) \rho^{-1} + \sum_{i=1}^n \|x_i^{(0)}\|^2. \quad (147)$$

Теорема 9 (о сильной состоятельности расширенного метода наименьших квадратов). Пусть последовательность оценок $\{\Theta_n\}$ строится в соответствии с проекционным алгоритмом (133). Пусть также

1) $\{\xi_n\}$ — последовательность независимых, одинаково распределенных случайных величин ξ_n , таких, что

$$E\{\xi_n\} = 0, \quad E\{\xi_n^2\} > 0, \quad E\{\xi_n^4\} < \infty \quad (n=1, 2, \dots);$$

2) функция $\varphi = \varphi(x)$ в (133) такова, что существует

$$\psi(x) \triangleq E\{\varphi(x + \xi_n)\} \quad \forall x \in R^1,$$

причем существуют положительные константы k_1, k_2, α, β , для которых

$$E\{\varphi^2(x + \xi_n)\} \leq k_1 + k_2 x \psi(x),$$

$$\alpha x^2 \leq x \psi(x) \leq \beta x^2,$$

$$\mu = \frac{\beta + \alpha}{2} \tilde{\lambda}_0 - \frac{\beta - \alpha}{2} \tilde{\lambda}_1 > \frac{1}{2},$$

$$\tilde{\lambda}_0 = \min \text{Re} G(e^{i\omega}), \quad \tilde{\lambda}_1 = \sum_{\omega=0}^{\infty} \left| \frac{1}{2\pi} \int_{-\infty}^{\infty} G(e^{-i\omega}) e^{ki\omega} d\omega \right|,$$

где $G(q) = [P_{\xi,1}^{-1}(q) - \frac{1}{2}]$ — SRP-оператор;

3) полином $Q(q) \triangleq \sum_{i=1}^N a_i^* q^i$ — устойчив, т. е. имеет корни вне единичной окружности на комплексной плоскости;

4) $\{u_n\}$ — последовательность случайных величин, таких, что с вероятностью 1

$$\overline{\lim}_{n \rightarrow \infty} \frac{1}{n} \sum_{i=1}^n u_i^2 \leq U^+(\omega) < \infty,$$

$$\overline{\lim}_{n \rightarrow \infty} \frac{1}{n^{1-\varepsilon}} \left(\sum_{i=1}^n u_i^4 \right) \leq u^+(\omega) < \infty, \quad \varepsilon \in (0, 1];$$

5) выполнено условие «достаточного разнообразия входов», т. е. для некоторого $\varepsilon \in (0, 1]$ с вероятностью 1

$$\lim_{n \rightarrow \infty} n^\varepsilon \lambda_{\min}(n^{-1} \Gamma_n^{-1}) \triangleq K_\lambda(\omega) > 0.$$

Тогда с вероятностью 1

$$\Theta_n \rightarrow \Theta \in C^*.$$

Покажем теперь, что рассмотренные алгоритмы асимптотически нормальны.

Теорема 10 (об асимптотической нормальности). Пусть выполнены условия теорем 7, 8, либо 9, и, кроме того, по вероятности

1) $n^\kappa \|n\Gamma_n - \Gamma\| \rightarrow 0$ при некотором $\kappa > 0$;

2) $n^\nu \left\| \frac{1}{n} \sum_{i=1}^n x_i^{(0)} v_i^T - R \right\| \rightarrow 0$ при некотором $\nu > 0$,

где v_i удовлетворяет уравнению $P_{\xi,1}(q) v_i = x_i^{(0)}$, $v_0 = x_0^{(0)}$, $v_{-m} = 0$ при $m > 0$;

3) $n^\eta \left\| \frac{1}{n} \sum_{i=1}^n x_i^{(0)} x_i^{(0)T} - X \right\| \rightarrow 0$ при некотором $\eta > 0$;

4) $\psi(\varepsilon)$ — дифференцируема в точке $\varepsilon = 0$, $\psi'(0) > 0$ и матрица $F = \left[\frac{1}{2} I - \psi'(0) \Gamma R \right]$ — устойчива.

Тогда

$$\sqrt{n} (\Theta_n - \Theta) \sim N(0, V),$$

причем матрица V является решением уравнения

$$FV + VF^T = -E \{ \varphi^2(\xi_1) \} \Gamma X \Gamma^T e^{F^T t} dt. \quad (148)$$

Уравнение (148) является уравнением Ляпунова, и в общем случае его решение представляется следующим образом:

$$V = E \{ \varphi^2(\xi_1) \} \int_0^{\infty} e^{Ft} \Gamma X \Gamma^T e^{F^T t} dt. \quad (149)$$

Так, для алгоритма стохастической аппроксимации (136)

$$\Gamma = \gamma_0 I, \quad F = \frac{1}{2} I - \psi'(0) \gamma_0 R, \quad \Gamma X \Gamma^T = \gamma_0^2 X,$$

для алгоритма Гудвина (137)

$$\Gamma = \gamma_0 (\text{tr } X)^{-1}, \quad F = \frac{1}{2} I - \psi'(0) \gamma_0 (\text{tr } X)^{-1} R,$$

$$\Gamma X \Gamma^T = \gamma_0^2 (\text{tr } X)^{-2} X,$$

для алгоритма расширенного МНК (138)

$$\Gamma = X^{-1}, \quad F = \frac{1}{2} I - \psi'(0) X^{-1} R, \quad \Gamma X \Gamma^T = X^{-1}.$$

ЗАКЛЮЧЕНИЕ

Оптимальные методы адаптивной идентификации основаны на учете априорной информации об объекте и помехах. Они дают возможность для заданного уровня априорной информации однозначно определить адаптивный предиктор, функцию потерь и, значит, рекуррентные алгоритмы. При этом часто оптимальные функции потерь отличны от квадратичных. Поэтому алгоритмы, как правило, нелинейны относительно невязки. Эти алгоритмы являются абсолютно оптимальными на классе в аргументном или критериальном смысле. Они отличаются грубостью, нечувствительностью к отклонениям от принятых при их формировании предположений. Иначе говоря, эти алгоритмы — робастные.

Для класса плотностей распределения с ограниченной дисперсией (класс P_2) робастными алгоритмами являются широко распространенные линейные относительно невязки алгоритмы, то есть алгоритмы ряда модификаций МНК.

Мы ограничились типовым, но простейшим описанием идентифицируемого объекта, устойчивым и минимально-фазовым. Нетрудно обобщить приведенные результаты на более сложные описания, на неустойчивые объекты (путем стабилизации их

обратной связью), на неминимально-фазовые объекты, на объекты с несколькими помехами. Робастные алгоритмы идентификации, несомненно, окажутся полезными и в задачах адаптивного управления. Они позволят создать адаптивные системы, оптимальные по скорости адаптации.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ ИНФОРМАЦИИ

1. Эйкофф П. Основы идентификации систем управления.— М.: Мир, 1975. 683 с.
2. Гроп Д. Методы идентификации систем.— М.: Мир, 1979.— 302 с.
3. Райбман Н. С., Чадеев В. М. Построение моделей процессов производства.— М.: Энергия, 1975.— 375 с.
4. Бокс Дж., Дженкинс Г. Анализ временных рядов. Прогноз и управление.— М.: Мир, 1979.— 406 с.
5. Goodwin G. C., Payne R. L. Dynamic system identification: experiment design and data analysis.— N. Y.: Academic Press, 1977.— 287 p.
6. Сейдж Э. П., Мелса Дж. Л. Идентификация систем управления.— М.: Наука, 1974.— 246 с.
7. Острем К. Введение в стохастическую теорию управления.— М.: Мир, 1973.— 321 с.
8. Саридис Дж. Самоорганизующиеся стохастические системы управления.— М.: Наука, 1980.— 402 с.
9. Ljung L., Söderström T. Theory and practice of recursive identification.— Cambridge, Mass.: The MIT Press, 1983.— 529 p.
10. Manczak K., Nahorski Z. Komputerowa idetyfikacja obiektów dynamicznych.— Warszawa: Państwowe wydawnictwo naukowe, 1983.— 363 p.
11. Gustavsson I. Comparison of different methods for identification of industrial processes // Automatica.— 1972.— 8, № 2.— P. 127—142.
12. Izerman R., Baur U., Bamberger W., Kueppo R., Siebert H. Comparison of six on-line identification and parameter estimation methods // Automatica.— 1974.— 10, № 1.— P. 81—103.
13. Saridis G. N. Comparison of Six on-line identification algorithms // Ibid.— P. 69—80.
14. Цыпкин Я. З. Синтез оптимальной настраиваемой модели в задачах идентификации // Автомат. и телемех.— 1981, № 12.— С. 62—67.
15. Цыпкин Я. З. Оптимальные критерии качества в задачах идентификации // Там же.— 1982, № 11.— С. 5—24.
16. Цыпкин Я. З. Оптимальные алгоритмы оценивания параметров в задачах идентификации // Там же.— № 12.— С. 9—23.
17. Tsyphkin Ya. Z. Optimality in identification of linear plants // Int. J. Syst. Sci.— 1983. 14, No. 1.— P. 59—74.
18. Цыпкин Я. З. Адаптация и обучение в автоматических системах.— М.: Наука, 1968.— 399 с.
19. Widrow B., Holt M. E. Adaptive switching circuits. IREWES CON Convention Record.— 1960.— Pt. 4.— P. 96—104.
20. Sacrison D. J. The use of stochastic approximation to solve the system identification problem // IEEE Trans. Autom. Contr.— 1967.— AC-12, № 5.— P. 563—567.
21. Saridis G. N., Stein G. Stochastic approximation algorithms for Linear discrete Time System Identification // Ibid.— 1968.— AC-13 № 5.— P. 515—523.
22. Kushner H. J., Clark D. S. Stochastic approximation methods for constrained and unconstrained systems.— N. Y.: Springer Verlag,— 1979.— 392 p.

23. *Plackett R. L.* Some theorems on least squares // *Biometrika*.— 1950.— 37, № 2.— P. 149—168.
24. *Peterka V.* A square root filter for real time multivariable regression // *Kibernetika*.— 1975.— 11, № 1.— P. 53—67
25. *Strejc V.* Least square parameter estimation // *Automatica*.— 1980.— 16, № 5.— P. 535—550.
26. *Mendel J. M.* Discrete technique of parameter estimation. The error formulation.— N. Y.: Dekker, 1937.— 453 p.
27. *Hsia T. S.* Identification: Least squares methods.— Lexington, mass.: Lexington books, 1977.— 261 p.
28. *Reiersl O.* Confluence analysis by means lag moments and other methods of confluence analysis // *Econometrica*.— 1941.— 9, № 1.— P. 1—23.
29. *Кендалл М., Стьюарт А.* Статистические выводы и связи.— М.: Наука.— 1973. 587 с.
30. *Wong K. Y., Polak E.* Identification of linear discrete time systems using instrumental variable approach // *IEEE Trans. Autom. Contr.*— 1967.— *AC-12*, № 6.— P. 707—718.
31. *Young P. C.* An instrumental variable method for real time identification of a noisy process // *Automatica*.— 1970.— 6, № 2.— P. 271—287.
32. *Young P. C.* Some observations on instrumental variable methods of time series analysis // *Int. J. Contr.*— 1976.— 23, № 5.— P. 593—612.
33. *Söderström T., Stoica P. G.* Instrumental variable methods for system identification.— Berlin.: Springer Verlag.— 1983.— 243 p.
34. *Landau I. D.* Unbiased recursive identification using model reference technique // *IEEE Trans. autom. contr.*— 1976.— *AC-21*, № 2.— P. 194—202; 1978.— *AC-23*, № 1. — P. 97—99.
35. *Feintuch P.* An adaptive recursive LMS filter // *Proc. IEEE*.— 1976.— 64, № 11.— P. 1622—1624.
36. *Мудров В. И., Кушко В. Л.* Методы обработки измерений.— М.: Сов. радио.— 1976.— 190 с.
37. *Кашьяп Р. Л., Рао А. Р.* Построение динамических стохастических моделей по экспериментальным данным.— М. Наука.— 1983.— 384 с.
38. *Sagara S., Wada K.* On-line modified least-squares parameter estimation of linear discrete dynamic systems. // *Int. J. Contr.*— 1977.— 25, № 3.— P. 329—344.
39. *Stoica P., Söderström T.* Bias correction in least-squares identification // *Ibid.*— 1982.— 35, № 3.— P. 449—457.
40. *Миддлтон Д.* Очерки теории связи.— М.: Сов. радио.— 1966.— 98 с.
41. *Сейдж Э., Мелса Дж.* Теория оценивания и ее применения в связи и управлении.— М.: Связь.— 1976.— 494 с
42. *Ван Трис Г.* Теория обнаружения, оценок и модуляции. Т. 1.— М.: Сов. радио.— 1972.— 744 с.
43. *Гильбо Е. П., Челпанов И. Б.* Обработка сигналов на основе упорядоченного выбора (мажоритарные и близкие к нему преобразования).— Там же.— 1976.— 189 с.
44. *Леман Э.* Проверка статистических гипотез.— М.: Наука. 1967.— 234 с.
45. *Харкевич А. А.* Борьба с помехами.— М.: Физматгиз.— 1963.— 197 с.
46. *Левин Б. Р.* Теоретические основы статистической радиотехники. Т. 2. М.: Сов. радио — 1975.— 392 с.
47. *Березин А В., Венцель В. А.* Теория и проектирование радиосистем.— Там же.— 1977.— 448 с.
48. *Пугачев В. С.* Теория случайных функций.— М.: Физматгиз.— 1962.— 884 с.
49. *Питерсон И. Л.* Статистический анализ и оптимизация систем автоматического управления.— М.: Сов. радио.— 1964.— 232 с.
50. *Вакман Д. Е., Седлецкий Р. М.* Вопросы синтеза радиолокационных сигналов.— Там же.— 1973.— 290 с.
51. *Беллман Р.* Процессы регулирования с адаптацией.— М.: Наука.— 1964. 360 с.
52. *Ljung L.* Convergence analysis of parametric identification methods // *IEEE Trans. Autom. Contr.*— 1978.— *AC-23*, № 11.— P. 770—783.
53. *Åström K. J.* Maximum likelihood and prediction error methods // *Automatica*.— 1980.— 16, № 5.— P. 551—574.
54. *Panuska V.* A stochastic approximation method for identification of linear systems using adaptive filtering.— Joint Automatic Control Conf. Michigan: Ann. Arbor, 1969.
55. *Panuska V.* Derivation of the Recursive Maximal Likelihood Identification Algorithm // *Electron. Lett.*— 1977.— 13, № 1.— P. 19—20.
56. *Kashyap R. L.* Estimation of parameters in partially whitened representation of a stochastic process / *IEEE Trans. Autom. Contr.*— 1974.— *AC-19*, № 1.— P. 13—21.
57. *Wittenmark B.* A Self-tuning Predictor // *Ibid.*— № 6.— P. 848—851.
58. *Goodwin G. C., Ramage P. J., Caines P. E.* A globally convergent adaptive predictor // *Automatica*.— 1981.— 17, № 1.— P. 135—140.
59. *Рао С. Р.* Линейные статистические методы и их применения.— М.: Наука.— 1968.— 547 с.
60. *Кульбак С* Теория информации и статистика.— Там же.— 1967.— 341 с.
61. *Алберт А.* Регрессия, псевдониверсия и рекуррентное оценивание.— Там же.— 1977.— 230 с.
62. *Цыкин Я. З., Поляк Б. Т.* Огрубленный метод максимального правдоподобия // *Динамика систем. Мат. методы теории колебаний.*— Горький, 1977.— Вып. 12.— С. 22—46.
63. *Huber P. J.* Robust estimation of location parameter // *Ann. Math. Statist.*— 1964.— 35, № 1.— P. 73—101.
64. *Tukey J. M.* A survey of sampling from contaminated distribution. Contribution to probability and statist. // Ed. by Olkin.— Stanford.: Stanford Univ. Press.— 1960.— P. 448—495
65. *Tukey J. W.* The future of data analysis // *Ann. Math. Statist.*— 1967.— 33, № 1.— P. 1—67.
66. *Huber P. J.* Robust statistics: a review // *Ibid.*— 1972.— 43, № 4.— P. 1041—1067.
67. *Hampel F. R.* Robust estimation: a condensed partial survey. // *Zeitschrift für Wahrscheinlichkeitstheorie und verwandte Gebiete.*— 1973.— 27.— P. 87—104.
68. *Bickel P. J.* Another Look at robustness: a review of reviews and some new development // *Scandinavian J. of Statist.*— 1976.— 3.— P. 145—168.
69. *Hampel F. R.* Modern Trends in the theory of robustness // *Math. Oper., Series Statist.*— 1978.— 9.— P. 425—442.
70. *Еришов А. А.* Стабильные методы оценки параметров // *Автомат. и телемех.*— 1978.— № 8.— С. 66—100.
71. *Andrews D. F., Bickel P. J., Hampel F. R., Huber P. J., Rogers W. H., Tukey J. W.* Robust estimation of location. Survey and advices.— Princeton: Princeton Univ. Press.— 1972.— P. 276.
72. *Хьюбер П.* Робастность в статистике.— М.: Мир.— 1984.— 304 с.
73. *Rey W. J.* Robust statistical methods.— Berlin: Springer Verlag, 1978.— 128 p
74. *Barnet V., Levis T.* Outliers in statistical data.— N. Y.: John Wiley and Sons,— 1978.— 350 p.
75. *Смоляк С. А., Титаренко Б. П.* Устойчивые методы оценивания.— М.: Статистика.— 1980.— 208 с.
76. *Robustness in statistics* / Eds. by *Launer R. L., Wilkinson G. N.*— N. Y.: Academic Press.— 1979.— 230 p.
77. *Robuste Verfahren* / Eds. by *Nowak H., Zeitgraf R.*— Berlin: Springer Verlag, 1980.— 472 p.
78. *Тихонов А. Н., Арсенин В. Я.* Методы решения некорректных задач.— М.: Наука, 1979.— 210 с.
79. *Морозов В. А.* Линейные и нелинейные некорректные задачи.— Итоги

науки и техники. Сер. Мат. анализ.— М.: ВИНТИ, 1973.— Т. 11.— С. 129—178.

80. *Martin R. D., Masreliez C. J.* Robust estimation via stochastic approximation // IEEE Trans. Inf. Theory.— 1975.— *IT-18*, № 3.— Р. 263—270.
81. *Martin R. D.* Robust estimation for time series autoregressions.— N. Y.: Academic Press.— 1979.— 200 p.
82. *Цыкин Я. З.* Основы информационной теории идентификации.— М.: Наука, 1984.— 320 с.
83. *Goodwin G. C., Payne R. I.* Dynamic system identification: Experimental design and data analysis.— N. Y.: academic press, 1977.— 287 p.
84. *Chen H. F.* Recursive estimation and control for stochastic systems.— N. Y.: John Wiley and Sons, 1985.— 378 p.
85. *Anderson B. D. O., Bitmead R., Kokotovic P. V., Kozut R. L., Mareels I. M. Y., Praly L., Reidle B. D.* Stability of adaptive systems. Passivity and analysis.— Cambridge, mass., London: The MIT Press, 1986.— 326 p.
86. *Деятериков И. П., Позняк А. С.* Оптимизация механических систем в условиях неопределенности: Учебное пособие.— М.: Изд-во МФТИ, 1984.— 88 с.

УДК 65.012.16

ПОЛУЧЕНИЕ ЭКСПЕРТНЫХ ЗНАНИЙ

И. В. Нейман, М. В. Шнейдерман

ВВЕДЕНИЕ

Работа посвящена анализу проблемы получения знаний специалистов, которая стала в последнее время предметом повышенного интереса. Он связан прежде всего с разработкой экспертных систем. Там ядро построения систем составляет процесс извлечения знаний из экспертов и преобразования их в программу. Общеизвестно, что получение знаний — наиболее принципиальный и в то же время сложный этап построения экспертной системы. Однако проблема получения знаний не ограничивается областью экспертных систем. Другой важной сферой являются методы сбора и анализа экспертной информации (экспертных оценок), т. е. методы проведения реальных экспертиз. Активное изучение этой области исторически предшествовало возникновению экспертных систем, но ее практическое значение остается высоким. В обзоре рассматриваются общие особенности, этапы, проблемы, трудности получения знаний. Более подробно излагаются методы получения знаний в двух названных областях: экспертные системы и организация экспертиз. Рассматриваются две группы методов: индивидуальное и коллективное получение знаний. В первом случае источником знаний служит один эксперт, во втором — экспертная группа. Обсуждаются вопросы связи методов получения знаний с формами организации знаний у человека.

1. ПРОБЛЕМА ПОЛУЧЕНИЯ ЭКСПЕРТНЫХ ЗНАНИЙ

Проблема получения экспертных знаний стала в последние годы предметом особого внимания, прежде всего в связи с созданием экспертных систем. Основу разработки экспертной системы составляет процесс получения знаний от экспертов и преобразование их в машинную программу. Общеизвестно, что получение знаний — центральный и в то же время наиболее сложный этап разработки экспертной системы.

Однако проблема получения знаний экспертов не ограничивается областью экспертных систем и имеет более долгую историю. Эта проблема возникала при различного рода консультационной деятельности, в задачах принятия решений, при проведении экспертных оценок тех или иных объектов. Активный период изучения проблемы продолжается около 30 лет (приблизительно с конца 50-х — начала 60-х годов).

В настоящее время можно выделить две основные сферы, где используются методы получения знаний: очная или реальная экспертиза и заочная экспертиза. Идея реальной экспертизы практически наиболее полно выразилась в организации экспертных оценок. При этом научную базу составили методы сбора и анализа экспертной информации [1—4]. Интенсивное изучение этой сферы исторически предшествовало развитию второй сферы и практическое значение ее остается весьма высоким.

Сфера заочной экспертизы реализуется в экспертных системах. Научную основу этой сферы составляют методы и средства искусственного интеллекта и экспертных систем [5—8], которые сейчас особенно интенсивно развиваются.

В работе [9] обозначенные выше сферы определены как человеческая компетентность и искусственная компетентность. Приведены аргументы в пользу последней (ее постоянство, легкость передачи и воспроизведения, невысокая стоимость эксплуатации).

В реальной жизни эти сферы взаимно дополняют одна другую. В качестве разделяющего их признака можно взять описание ситуации, в которой требуется консультация специалистов. Экспертные системы ориентированы на стандартные описания ситуаций в той или иной предметной области. Например, в области медицинской диагностики все пациенты описываются заданным общим списком симптомов, а каждый пациент имеет свой набор симптомов. Система заранее наполняется знаниями о симптомах и способах вывода диагностических заключений. Каждому конкретному пациенту система дает заочную консультацию (на основе этих знаний) с экспертами, передавшими свой опыт системе.

Реальные экспертизы в основном ориентированы на ситуации, не поддающиеся стандартному описанию, уникальные,

особо важные и т. п. В медицине это может быть пациент с не встречавшимся раньше симптомом, с редким сочетанием симптомов, особо тяжелый больной и т. п. Понятно, что в этих случаях нужна очная и, как правило, оперативная консультация с врачами высокой квалификации.

Экспертные системы охватывают сейчас все новые и новые области и задачи, поэтому четкое разделение указанных сфер еще не оформилось, однако заведомо ясно, что обе сохраняют практическое значение. Специфика выделенных сфер будет пояснена ниже, сейчас же попытаемся остановиться на общих особенностях получения знаний.

Приобретением знаний назовем процесс получения (извлечения) знаний от одного или нескольких экспертов и представления знаний в форме, обусловленной решаемой задачей. Существенны два момента этого процесса: получение знаний и их представление. С одной стороны, источник знаний — эксперт обладает обширными и обычно слабо структурированными знаниями (фактами, эвристиками, правилами, мнениями и т. п.), с другой, решаемая задача обуславливает специфическую, нередко весьма непривычную для экспертов форму представления.

Поэтому между экспертами и системой, решающей задачу (экспертной системой, механизмом экспертизы), существует посредник. В экспертных системах — это инженер по знаниям, в механизмах реальной экспертизы — исследователь, организатор экспертизы. Его задача — преодолеть несоответствие между «внутренним» содержанием знаний экспертов и «внешней» формой их представления, помочь экспертам сформулировать их суждения в требуемой форме. Процесс приобретения знаний изображен на рис. 1.

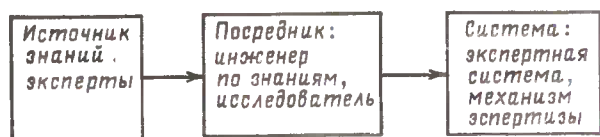


Рис. 1

Эксперты обладают в основном знаниями о предметной области и мало знают об особенностях системы. Посредник, в свою очередь, меньше знает область и больше — способы приобретения знаний. Поэтому очень важно взаимодействие источника и посредника.

Формы представления знаний в экспертных системах и реальных экспертизах весьма различны. В первом случае — это правила продукций, фреймы, семантические сети и другие формы, допускающие их воплощение в виде программы. Во втором случае — это оценочные суждения, как правило, в виде

оценок на шкалах измерений. В любом случае важно разделять методы получения и формы представления знаний. Расхождение между ними составляет главный узел проблем приобретения и использования знаний.

Литература по проблеме представления знаний существенно объемней, чем литература по проблеме получения знаний. Это объясняется, видимо, тем, что проблема получения знаний значительно сложнее, в частности, потому, что затрагивает тонкие психологические вопросы взаимоотношений эксперта и посредника.

Получение знаний — комплексная проблема, имеющая технические, экономические, психологические, социологические и иные аспекты. Роль знаний и информации в современной экономике освещена в [10]. Социокультурные аспекты получения и передачи знаний в человеческом обществе затронуты в [11]. Там же подчеркнута, что только интеграция усилий специалистов различных дисциплин позволит создать всесторонне обоснованные, эффективные системы получения и использования знаний.

Особое внимание уделялось получению знаний психологами. Природа, психологические особенности и ограничения сбора экспертных знаний показаны в [12, 13]. Целый ряд работ, главным образом психологов, позволил выявить основные трудности в получении знаний.

Первая трудность состоит в расхождении между способами, которыми эксперты обычно выражают свои знания, и формами, в которых знания могут быть представлены в системе [7]. Есть целый ряд путей уменьшения этого расхождения: и разработка соответствующих методов получения знаний, и активизация роли посредника, и совершенствование форм представления, — последний очень актуален. Речь идет о выработке форм, приближенных к тем естественным для специалиста формам, которые он использует в повседневной практике.

В [14] предполагается, что экспертные системы выиграли бы оттого, что в них заложат такой же тип представления знаний, какой связан с человеческим опытом. Имеются попытки работ в этом направлении. Так, при построении экспертных систем пытаются использовать формы представления знаний, которые допускают эквивалентное представление на естественном языке. Такой подход, например, был применен при создании системы ROSIE [7]. Что касается преодоления первой трудности на путях получения знаний, то об этом будет сказано при описании методов получения знаний.

Вторая трудность заключается в неспособности эксперта вербализовать, ясно выразить словами свои знания. Отмечено, что очень часто эксперт затрудняется описать знания в точной, понятной, связной форме. Эта трудность объясняется подсознательным характером многих знаний человека и сложностью их выявления [7, 11]. Характерно, что это свойство присуще не

только экспертной, но и многим другим областям деятельности человека. Получаемые вербальные отчеты экспертов мало отражают ход «внутренних» рассуждений [15].

По мере накопления опыта специалист нередко все более утрачивает умение словесно выразить свои знания, что связано с трудностью описания мыслительных процессов. Часто эксперт оказывается неспособным объяснить, какие эвристические приемы он использовал при решении конкретной задачи, и эта неспособность к вербализации возрастает с ростом опыта и компетентности специалиста. Например, при обучении вождению автомобиля новичок старается в точности следовать правилам вождения. Чем выше уровень, тем больше человек не столько следует правилам, сколько комбинирует из них различные планы действий, соответствующие условиям езды. Наконец, эксперт управляет машиной инстинктивно, исходя не из правил, а из опыта, накопленного в разных ситуациях. При этом эксперт затрудняется назвать правила, которые он когда-то изучил. Сказанное справедливо и для многих других профессий, в особенности профессий, связанных с психомоторными действиями (спортивные игры, управление самолетом, игра на музыкальных инструментах). По-видимому, правила, усвоенные экспертом при обучении, постепенно образуют сложную и индивидуализированную структуру. Таким образом, неспособность к вербализации — не дефект, а результат структурной перестройки организации знаний в памяти эксперта [16].

К этой трудности добавляется еще одна: если даже эксперту удалось как-то выразить свои знания, они могут оказаться недоступными, непонятными людям, выходящим за круг экспертов, напр. новичкам. Дело в том, что последние не владеют узкопрофессиональным языком экспертов, а значит не смогут воспользоваться полученными знаниями [11].

Третья трудность связана с тем, что при сообщении своих знаний человек постоянно проявляет т. н. смещения (bias) — систематические отклонения субъективных суждений от суждений, соответствующих моделям рационального поведения. Накоплено достаточно результатов экспериментальных психологических исследований, показывающих, что в процессе решения задач человек ведет себя непоследовательно, противоречиво и т. д. [17, 18]. Так, при сравнении многокритериальных альтернатив в поведении людей наблюдается нетранзитивность, непоследовательность, противоречия, стремление к резкому упрощению задачи.

Важный вид смещений связан с влиянием на суждения и решения человека априорных знаний [19]. Априорные знания играют большую роль в экспертных решениях, но они часто являются источником предвзятости, неосознанного стремления подтвердить уже имевшиеся убеждения, что может пойти в

ущерб искомому решению. Люди часто склонны переоценивать или недооценивать вероятности событий, проявляя при этом самоуверенность. Излишнюю самоуверенность люди демонстрируют и относительно точности назначаемых оценок каких-либо параметров. Дополнительным источником погрешностей может быть не только эксперт, но и посредник (инженер по знаниям) [20].

Итак, установлено что экспертные знания и суждения подвержены влиянию разного рода смещений. Это порождает очень серьезные проблемы получения и использования знаний. Экспертные системы не должны механически копировать действия эксперта при решении задач. Системы должны брать те компоненты экспертного мышления, которые воплощают интеллект, и отсекают компоненты, несущие смещения. Легче отсечь эти компоненты в процессе получения знаний, чем в процессе корректировки уже построенной базы знаний.

Однако это не значит, что в процессе получения знаний легко избавиться от смещений. Соответствующие методы получения знаний нельзя разработать до тех пор, пока не будут четко выделены все виды смещений и преодолены трудности вербализации знаний [21]. Психологические исследования могут служить лишь ориентирами, а прямые выводы о роли смещений в получении экспертных знаний должны быть получены путем специальных экспериментов.

Четвертая трудность состоит в том, что экспертные суждения могут быть несовершенными: неполными, неточными, не относящимися к поставленному вопросу и т. д. [11]. Здесь может играть роль круг причин — зависимость суждений от конкретной ситуации, условий опроса, искажение (часто неосознанное) экспертами их действительных мнений, несовершенство самих экспертов. По-видимому, преодоление этой трудности возможно на пути улучшения организации экспертизы — подбора экспертов, их инструктажа, выработки процедуры работы с экспертами и т. п. (см., напр., [4]).

2. ОСНОВНЫЕ ОБЛАСТИ ПОЛУЧЕНИЯ ЗНАНИЙ

Экспертные системы. Как уже говорилось, получение знаний — часть более общего процесса приобретения знаний. Последний в экспертных системах определяется как процесс получения знаний от эксперта (или иного источника) и преобразования их в программу [7]. Этот процесс идет параллельно со всем жизненным циклом экспертной системы: от начала разработки до создания действующего образца. Обычно экспертные системы строятся итеративно: после начальной стадии создается прототип, который затем углубляется и расширяется.

Формализация предполагает перевод ключевых понятий, отношений, подзадач, выделенных на предыдущей стадии, в неформальные представления, основанные на методах и средствах инженерии знаний. Здесь активную роль играет владеющий этими методами инженер по знаниям. Он должен выбрать форму и язык представления знаний и с помощью эксперта выразить на нем понятия и отношения.

На стадии реализации инженер по знаниям и эксперт формулируют множество правил, воплощающих знания. Эти правила соответствуют выбранной на предыдущей стадии схеме представления. Множество правил и соответствующих управляющих структур определяет программу-прототип, которую можно выполнять и подвергать контрольным испытаниям. При создании прототипа используются различные программные средства, включая текстовые редакторы и интерфейсы для формирования баз знаний.

На последней стадии проводятся испытания прототипа на нескольких примерах с целью его приведения в соответствие с принятыми стандартами качества решений в данной области. Эту работу выполняет инженер по знаниям, обращающийся к эксперту для оценки работы программы и консультаций по модификации прототипа.

Делаются попытки автоматизации приобретения знаний в целях переложить ответственность за выполнение перечисленных стадий с человека на ЭВМ. Пока что эти попытки весьма ограничены. В очень малом числе систем процесс приобретения знаний хотя бы частично автоматизирован. В них автоматизируются в основном две последние стадии, а более ранние выполняются ручными методами.

Активная роль эксперта как средства получения знаний проявляется на стадиях идентификации, концептуализации и реализации. Здесь проводится совместная работа эксперта и инженера по знаниям по получению от специалиста и вводу в систему знаний о предметной области.

В сегодняшней практике экспертных систем чаще всего взаимодействуют один эксперт и один инженер. Эксперт выступает как информатор, рассказывающий о своем опыте, не обязательно располагая продуманной стратегией обучения. Инженер по знаниям получает, многократно переформулирует знания и проверяет их, используя тестовые примеры [7].

В принципе могут и должны привлекаться несколько экспертов. Практика коллективных экспертиз доказала преимущества коллектива специалистов при решении разнообразных задач. При этом всегда возникали сложности, связанные с наличием несовпадающих точек зрения. В методах экспертных оценок эти сложности преодолеваются развитым аппаратом выявления и анализа различных мнений, их согласования и взаимообогащения, выявления преобладающих мнений [3, 4].

В области экспертных систем эти сложности пока трудно преодолимы, хотя в будущем разработчикам неминуемо придется ориентироваться на учет знаний нескольких экспертов. Из более или менее реальных для экспертных систем подходов можно сегодня указать т. н. персонификацию знаний [8]. Она состоит в том, что каждое правило можно помечать именем эксперта — автора этого правила. Такой подход дает возможность выяснять, как решает тот или иной вопрос такой-то эксперт, сравнивать рассуждения различных экспертов. Однако изучение этих проблем только начинается.

Итак, в настоящее время получение знаний происходит в результате совместной работы эксперта и инженера по знаниям. Причем первый выступает в качестве источника информации, второй — ее получателя. Однако отсюда не следует, что эта совместная работа не может подкрепляться соответствующими методами и средствами.

Содержание работы инженера по знаниям и эксперта, т. е. задаваемые вопросы и ответы на них, чрезвычайно разнообразно, зависит от формы представления, стадии разработки системы. Отметим еще раз, что эксперт высказывается на своем профессиональном языке, а инженер задает вопросы и использует высказывания с тем, чтобы реализовать определенные формы представления.

На начальной стадии развития экспертных систем проблеме получения знаний не придавалось должного значения. Сейчас эта проблема, как и проблема представления знаний, признается центральной. Самое совершенное программное обеспечение экспертной системы будет бесполезным при недостаточном внимании к получению знаний [19].

Далее мы подробно остановимся на методах получения знаний. Под ними понимаются общие методы работы с экспертами с целью извлечения их знаний, т. е. общий инструментарий, из которого можно выбирать тот или иной метод для решения данной задачи. Речь идет о достаточно универсальных способах работы с экспертами, базирующихся на результатах психологических и социально-психологических исследований. Эти методы могут использоваться как при построении экспертных систем, так и при организации экспертиз. Но прежде, чем перейти к их изложению, рассмотрим вторую из двух названных сфер.

Организация экспертиз. Если в экспертных системах получение знаний нужно для построения системы, которая далее сама выдает заключения, то в реальных экспертизах ситуация иная — эксперты привлекаются каждый раз заново для решения данной задачи. Отсюда исходят и существенные различия в подходе к получению знаний.

Первое отличие состоит в характере и продолжительности процедуры получения знаний в экспертных системах, как го-

ворилось выше, приобретение знаний — длительный процесс, тесно связанный со всем жизненным циклом системы, состоящий из целого ряда стадий и длящийся часто по несколько месяцев. Это понятно — база знаний должна быть построена очень тщательно, т. к. потом она может использоваться многие годы. В реальных экспертизах решается «одноразовая» задача; эксперты привлекаются, как правило, на непродолжительное время: их опрос отнимает максимум несколько часов, а при многотуровом опросе — несколько раз (обычно 2—4 раза) по несколько часов.

Второе отличие состоит в числе работающих экспертов. При построении экспертных систем чаще всего источником знаний служит один специалист, с которым тщательно работает инженер по знаниям. Использование нескольких экспертов при сложном и длительном процессе построения системы в настоящее время затруднительно. В подавляющем числе реальных экспертиз участвует коллектив специалистов. Такая апелляция к коллективной мудрости берет истоки из глубин истории человечества — примером тому военные советы, советы старейшин, коллегии и т. д. Коллективность экспертизы заставляет делать акцент на процедурах совместного опроса, сопоставлении различных точек зрения, их математического анализа.

Третье отличие заключается в роли посредника. Инженер по знаниям при разработке экспертной системы играет исключительно активную роль, поскольку здесь велико значение методов инженерии знаний, которыми владеет инженер и не владеет эксперт. Задача инженера — вести активный и продолжительный диалог с экспертом. В реальных экспертизах на исследователя выпадает большой объем работ по подготовке и проведению опроса, обработке результатов и т. п. На этапе получения знаний (т. е. экспертного опроса) задача исследователя сводится в основном к тому, чтобы подготовить и организовать совместную работу членов экспертной группы.

Обычно процесс реальной экспертизы состоит из ряда этапов: подготовка экспертизы (формулировка проблемы, выделение критериев, построение шкал измерений и др.), отбор экспертов и формирование экспертной группы, опрос экспертов, анализ и обработка результатов опроса. Нас интересует этап получения знаний — экспертный опрос.

Содержание работы исследователя и экспертов, т. е. вопросы и ответы на них, зависит от формы представления, диктуемой решаемой задачей. Форма представления в реальных экспертизах традиционно связывается с основными типами шкал измерений: шкалы наименований, порядка, интервалов, отношений. Получаемые от экспертов формы (суждения) или прямо представляют собой оценки по этим шкалам, или преобразуются в эти оценки (напр., из парных сравнений). Наиболее употребительны на практике суждения следующих видов:

парные сравнения, множественные сравнения, ранжирование, классификация, непосредственная шкальная оценка (по шкалам интервалов или отношений) [2, 4, 23]. Кроме этих традиционных форм иногда используются и более сложные: структуры сложных объектов, деревья целей, развернутые текстовые сценарии. В любом случае здесь, как и в экспертных системах, также не снимается проблема разрыва между естественным для специалиста языком выражения знаний и формой их представления.

Здесь также предпринимаются попытки решения этой проблемы. Одна из них состоит в том, что от экспертов требуют не только суждение в заданной форме (обычно числовой оценки), но содержательное его обоснование на профессиональном языке (аргументы, мнения, сведения) (см. в [4]).

Проблема получения экспертных знаний на ранних этапах развития методов экспертных оценок недооценивалась так же, как и в начале развития экспертных систем. Недооценка этой проблемы при проведении экспертизы ставит под сомнение ценность ее результатов. Между тем такая недооценка встречается и сегодня, что выражается в непродуманном, поспешном опросе специалистов. Низкое качество собранных таким образом мнений не может быть скомпенсировано использованием для их обработки современных математических методов. В [4] показано значение названной проблемы и, что существенно, ее отличие от проблемы массового опроса населения, т. е. сбора неэкспертных мнений.

Теперь перейдем к рассмотрению методов получения знаний. Как уже отмечалось, это общие методы работы с экспертами, которые в принципе могут использоваться как при построении экспертных систем, так и при проведении экспертных опросов.

3. ИНДИВИДУАЛЬНОЕ ПОЛУЧЕНИЕ ЗНАНИЙ

Методы получения знаний разделим на две группы: индивидуального и коллективного получения знаний. В первом случае источником знаний является один эксперт, во втором — экспертная группа. Индивидуальные методы имеют двоякий практический смысл. Во-первых, они применяются тогда, когда эксперт действительно один (как это часто бывает при построении экспертных систем). Во-вторых, эти методы применяются и при коллективном получении знаний. Далее будет показано, что методы коллективного получения знаний делятся на методы, предусматривающие непосредственное общение экспертов, и методы, исключающие такое общение. В последнем случае на каждом туре многотурового опроса эксперты самостоятельно сообщают знания (после чего происходит об-

мен знаниями). Здесь и могут применяться методы индивидуального получения знаний.

Далее предполагается, что рассматриваются общие методы, не ориентированные на какие-то predetermined формы представления знаний. Общепризнанно (напр., [24]), что основные методы — это интервьюирование и анкетирование.

Интервьюирование — это беседа между экспертом и исследователем, в которой последний задает вопросы в соответствии с более или менее разработанной программой. Особенность интервью в том, что эксперт не имеет много времени для глубокого продумывания ответов, а исследователь строит вопросы в значительной степени в зависимости от ответов эксперта на предыдущие вопросы [2].

Исследователь должен всесторонне изучить анализируемую проблему, уметь ясно формулировать вопросы, создавать непринужденную обстановку, слушать собеседника. Подготовка исследователя к беседе заключается в основном в разработке цели интервью, подготовке главных вопросов, суть которых должна быть записана, хотя окончательные формулировки могут производиться в ходе разговора [2].

В [25] выделены следующие формы интервью: свободная беседа, беседа «вопрос — ответ», «перекрестный допрос». Первый способ отличается от второго меньшей выраженностью системы вопросов. Третий осуществляется с привлечением нескольких интервьюеров по принципу перекрестного допроса в судебном расследовании. Интервьюеры стремятся получить максимум информации, проверяя в то же время непротиворечивость, последовательность, логичность рассуждений эксперта.

В [26] дан ряд рекомендаций исследователю для достижения эффективности интервью.

1. Обеспечение взаимопонимания с экспертом. Для этого следует учитывать возможные препятствия к этому со стороны эксперта. Например, эксперт затрудняется выразить свои мысли из-за сложности проблемы, но чувствует, что от него ожидают ясности изложения. Необходимо снимать такого рода затруднения.

2. Необходимость формулирования вопросов в более свободной форме, а затем постепенно их конкретизация, с концентрацией внимания на узловых моментах. При этом эксперт вырабатывает свой «словарь» и втягивается далее в серьезный анализ проблемы.

3. Недопустимость навязывания эксперту своего понимания проблемы. Необходимость выслушивания его аргументов, даже если они кажутся не относящимися к делу. При непонимании эксперта попытайтесь выявить его понимание предмета, а не утвердить свое.

4. Лимитирование времени интервью с учетом ограниченных возможностей эксперта, связанных с утомлением и т. п.

причинами, но без обрывания беседы на важном месте, в процессе активного размышления эксперта.

В [26] отмечается, что интервью — наилучший способ получения знаний на ранних стадиях, когда проводится выявление основных объектов, связей между ними и т. д. Полезные инструкции по организации интервью приводятся также в [27].

Достоинства и недостатки интервьюирования перечислены в [2]. Достоинством является непрерывный живой контакт интервьюера и опрашиваемого, что позволяет быстро получить большое количество информации и всесторонне, хотя и возможно поверхностно, осветить предмет экспертизы. К недостаткам относятся возможность сильного влияния интервьюера на ответы эксперта, отсутствие у последнего времени на глубокое продумывание ответов, высокие требования к интервьюеру, значительные затраты времени на опрос.

Анкетирование заключается в предъявлении эксперту анкеты, на вопросы которой он должен дать ответы в письменной форме. Конкретный вид анкеты и содержание вопросов определяются спецификой проблемы.

Анкетирование делая на очное и заочное. В первом случае заполнение анкеты проводится экспертом в присутствии исследователя. При этом, однако, возможно влияние последнего на ответы эксперта. При заочном способе прямого контакта с исследователем нет, эксперт работает с анкетой один. Достоинства заочного способа — простота его организации, возможность привлечения далеко живущих экспертов (анкета посылается по почте) Но при этом возможны неправильное истолкование вопросов, задержки с ответом и т. п.

Вопросы анкеты можно разделить на три группы [28]: 1) объективные данные о самом эксперте, 2) основные вопросы по существу анализируемой проблемы, 3) дополнительные вопросы по источникам информации и аргументации эксперта.

По форме основные вопросы делятся на открытые и закрытые. Ответ на открытый вопрос может быть дан в произвольной форме. Закрытый вопрос сопровождается вариантами (веером) возможных ответов, и эксперт должен остановить свой выбор на одном (иногда нескольких) из них. Преимущество открытых вопросов в возможности выявления новых, заранее не известных аспектов проблемы, а недостатки — в произвольной интерпретации вопросов и в сложности обработки ответов. Наоборот, закрытые вопросы однозначно интерпретируются, требуют меньше времени на ответы, причем последние легче обрабатывать. Вместе с тем появляется опасность навязывания эксперту ответов, особенно в тех случаях, когда он не имеет сложившегося мнения или когда оно не укладывается в веер ответов. Поэтому следует делать веер широким и предоставлять эксперту возможность выйти за его пределы [2, 28].

В ряде случаев кроме основных прямых вопросов применя-

ются косвенные. Их цель может состоять в получении информации косвенным образом, когда известно, что эксперт по каким-то причинам может уклониться от ответа на прямой вопрос, или в проверке компетентности эксперта, его отношения к опросу и т. п. [2].

В [1] обращается внимание на формулировку вопросов и порядок их постановки. Вопросы не должны допускать неоднозначного толкования. Сначала лучше требовать ответы на более общие вопросы, а затем переходить к более частным. Набор вопросов должен быть логически взаимосвязан.

По сравнению с интервьюированием анкетирование — более эффективное и оперативное средство сбора знаний. При этом эксперт находится в более спокойной, комфортной обстановке. Но анкетирование дает меньше возможностей для генерирования непредвиденных данных, для более полного и свободного раскрытия творческого потенциала эксперта и более пригодно для получения нечетких и вероятностных суждений [26].

Рассмотрев главные, остановимся на остальных методах индивидуального получения знаний.

Наблюдение за работой эксперта, по мнению некоторых авторов, это лучший способ узнать, как эксперт вырабатывает решения. Он состоит в наблюдении за тем, как эксперт работает в реальных условиях. Первая задача, стоящая здесь перед исследователем, — найти способ регистрации действий эксперта. Самый легкий путь — простое наблюдение и запись действий эксперта с одновременной попыткой их анализа. Более сложные способы связаны с применением технических средств, например видеозаписи. Тогда анализ наблюдений проводится позднее, в спокойной обстановке, возможно, при помощи самого эксперта [26].

Анализ протоколов — это фактически усовершенствованная форма предыдущего способа. Как и там, эксперт решает реальную задачу. Но теперь исследователь просит эксперта в ходе выполнения задачи «мыслить вслух». Исследователь может дать инструкцию типа: «думайте о целях, формулируйте метод поиска решения» и т. д. Запись такой работы служит далее объектом анализа [9, 26].

Данный способ ограничивается теми видами задач, для которых вербализация является естественной частью общего процесса рассуждения. Только тогда высказанное вслух может рассматриваться как более или менее надежный источник информации. Для целого ряда задач этот способ неприемлем и процесс «мышления вслух» может создать искаженное представление о реальных механизмах решения задач — это задачи, для которых не естественна вербализация рассуждения. Типичный пример — задачи, связанные двигательной реакцией человека в ответ на внешний стимул [29].

Обычно при анализе протоколов стараются выявить основные типы объектов, которыми оперирует эксперт, взаимосвязи между ними и типы заключений, используемые экспертом. Примеры анализа можно найти в [30].

Анализ прерываний — метод, допускающий, что эксперту не надо «мыслить вслух». Однако когда в процессе решения задачи экспертом наступает такой момент, что исследователь теряет нить действий эксперта, исследователь прерывает его и просит объяснить ход работы [26]. Далее работа эксперта продолжается. Сложность состоит в том, что эксперт, будучи прерван, с трудом начинает работать дальше.

Применительно к экспертным системам данный метод наиболее ценен, когда уже имеется прототип системы и действия эксперта можно сопоставлять с работой системы.

В заключение отметим, что индивидуальные методы мало поддаются автоматизации и остаются в основном в рамках традиционной неформальной работы исследователя и эксперта. Программные средства для приобретения знаний развиваются в области экспертных систем [7]. Но они главным образом затрагивают вторую часть этого процесса — представление знаний в системе, меньше касаясь получения знаний от экспертов. Обзор названных программных средств выходит за рамки данной работы. Это самостоятельная область, требующая специального рассмотрения, которое проводится в [7, 31] и многих других работах.

4. КОЛЛЕКТИВНОЕ ПОЛУЧЕНИЕ ЗНАНИЙ

Методы прямого взаимодействия. Коллективное получение знаний имеет исторические корни и широко используется при проведении реальных экспертиз. В будущем оно, по-видимому, будет использоваться и при построении экспертных систем. В этом случае, возможно, существующие коллективные методы подвергнутся модификации.

При изучении существующих методов коллективного получения знаний выделяются [4] два ведущих фактора: характер взаимодействия экспертов в ходе опроса и характер «обратной связи», т. е. информирования экспертов в данный момент опроса о его предыдущем ходе. По первому фактору можно указать: а) процедуры, предусматривающие непосредственное личное взаимодействие экспертов, б) процедуры, исключающие такое взаимодействие. По второму фактору выделяются: а) итеративные (многотуровые) процедуры с явной обратной связью между турами, б) простые однотуровые процедуры. В итоге можно назвать четыре типа методов: 1) однотуровые с непосредственным взаимодействием, 2) однотуровые без непосредственного взаимодействия, 3) многотуровые с непосред-

ственным взаимодействием, 4) многотуровые без непосредственного взаимодействия.

В данной работе мы воспользуемся более простой классификацией. Во-первых, практика показала, что однотуровые методы без взаимодействия не отвечают специфике экспертного опроса, а ориентированы на массовый опрос населения. Во-вторых, фактор вида взаимодействия оказывается столь существенным, что позволяет объединить методы первого и третьего типа. Поэтому мы возьмем за основу два класса методов: процедуры с прямым взаимодействием и интеративные процедуры с косвенным взаимодействием.

Образцом первого класса является традиционная «дискуссия за круглым столом». В ходе дискуссии эксперт имеет возможность неоднократно высказывать суждения, учитывая точки зрения других участников опроса. В традиционной дискуссии фактически действует перманентная и неконтролируемая обратная связь, по которой эксперт получает мнения других экспертов, а также их более общую реакцию (в том числе невербальную).

Методы второго класса ведут свою историю от широко известного метода Делфи. Здесь эксперты изолированы друг от друга, а процедура реализуется за несколько разделенных во времени туров (итераций). На каждом туре эксперт под контролем исследователя получает по обратной связи информацию о суждениях других членов группы (обезличенно) и пересматривает свое суждение.

На практике шире применяются методы первого класса. Основная причина в их более легкой и простой реализации. В то же время теоретический приоритет в последнее время отдается методам второго класса. Но их осуществление связано с большими затратами времени и организационными усилиями. Однако эти усилия компенсируются существенными достоинствами методов второго класса, связанными со снижением уровня конформизма, давления на экспертов мнения большинства, присущего традиционной дискуссии.

Оба класса методов, видимо, не потеряют свое значение и в будущем. Выбор одного из них, так же как и выбор конкретной процедуры, определяется целью и характером экспертизы, сущностью решаемой проблемы, лимитами времени, доступностью экспертов и т. д.

Начнем изложение с методов первого класса. Здесь на первом месте стоит дискуссия за круглым столом, очень давно и широко используемая в практике экспертных опросов. Нередко дискуссия проводится с перерывами в несколько четко выраженных туров [4]. Дискуссия, как правило, включает в себя вступительное слово ведущего, доклад по анализируемой проблеме, вопросы и ответы, выступления экспертов, проведение итогов. Устный доклад может и отсутствовать, если

участникам розданы письменные отчеты о предварительном анализе проблемы. Типичное выступление эксперта состоит из изложения точки зрения с выделением моментов согласия и расхождения с оппонентами, аргументации своей позиции, выводов и предложений [2].

В дискуссии очень важна роль ведущего, задача которого — всемерно способствовать всестороннему анализу проблемы экспертами и в то же время препятствовать принятию компромиссного решения, не отвечающего интересам дела. Рекомендации на этот счет содержатся в [2].

К достоинствам дискуссии относятся [32]: оперативная обратная связь, наличие невербальных контактов, быстрое преодоление непонимания (например, терминологического) между экспертами. Основные недостатки: сильное влияние суждений крупных авторитетов, подверженность участников мнению большинства, нежелание открыто пересмотреть уже высказанное суждение.

Психологи, изучавшие поведение участников дискуссии, обнаружили опасность возникновения явления «группового мышления». Оно возникает, когда, несмотря на высокий интеллект членов группы, они отказываются от критического, реалистического стиля мышления во имя единомыслия, основанного на излишней убежденности в успехе групповой деятельности [33, 34].

Другой известный метод — мозговая атака, состоящая в проведении совместного заседания экспертов по определенным правилам, направленным на создание атмосферы свободного высказывания суждений [35, 36]. Принципиальным является отказ от критической оценки высказанных во время заседаний суждений; в этом смысле механизм обратной связи здесь отсутствует. Метод преследует решение двух задач: генерирование новых идей, анализ и оценка ранее предложенных. Правила проведения заседания перечислены в [2]. Здесь еще более велика роль ведущего, от искусства которого существенным образом зависит успех мозговой атаки. Есть ряд разновидностей мозговой атаки — метод отнесенной оценки [37] и т. д. (см. [2]).

Мозговую атаку можно считать весьма специфичной процедурой опроса, пригодной в случаях, требующих творческой атмосферы и максимального разнообразия суждений. Это случай дефицита творческих решений, новых идей, свежих концепций [2, 4].

Своеобразным является метод номинальной группы [38]. Эксперты сначала в присутствии друг друга, но никак не общаясь, формируют свои суждения, которые представляются для всеобщего обозрения. После того, как все суждения будут зафиксированы, начинается общее открытое их обсуждение. Оно заканчивается голосованием, в результате которого выбирается суждение, поддерживаемое большинством. Делается попытка разделить этапы выдвижения идей и их обсуждения, при-

чем на первом этапе считается важной самостоятельная независимая работа экспертов. Это стимулирует к индивидуальному творчеству, уменьшает вероятность спешных решений и т. п. По своей направленности метод близок к мозговой атаке на стадии выдвижения мнений

Имеется немало и других разновидностей названных методов (см., например, [2]). Сюда же в какой-то степени примыкают такие способы групповой работы, как деловые игры, сценарии и т. п. [24]. Но в любом случае неизбежной остается основная черта этих методов — контакты экспертов лицом к лицу. Именно с ней связаны существенные дефекты, присущие дискуссии за круглым столом, но проявляющиеся так или иначе во всех методах этого класса.

Помимо этих недостатков имеются и сопутствующие им трудности организационного характера, связанные с необходимостью собрать одновременно в одном месте ведущих в какой-то области знаний специалистов. К тому же не всегда удается в результате одного заседания получить приемлемый результат [24].

Методы второго класса как раз и преследуют цель преодоления всех этих недостатков.

Методы косвенного взаимодействия. Во всех методах второго класса эксперты изолированы друг от друга; при наличии соответствующих средств коммуникации они могут быть на большом отдалении как один от другого, так и от исследователя. Естественно, что в таких условиях одноразовое «включение» какого-то эксперта будет неэффективным. Экспертный опрос здесь осуществляется в несколько туров. На каждом из них каждый участник получает посредством исследователя информацию, характеризующую позицию других экспертов, под воздействием которой может скорректировать свое мнение. В перерывах между турами исследователь анализирует собранные мнения и готовит материал для адресации экспертам. Иначе говоря, имеет место контролируемая исследователем обратная связь, организация которой существенно отличает один метод от другого.

Как вы видим, методы этого класса предполагают, что на каждом туре каждый эксперт самостоятельно формирует суждения. Здесь и могут использоваться методы индивидуального получения знаний, о чем говорилось выше. На практике в большинстве случаев здесь применяется метод анкетирования.

Первым в ряду этих методов идет метод Делфи [39, 40], многократно описанный в литературе. Напомним, что экспертам предъявляется оцениваемый объект (один или несколько). Опрос экспертов осуществляется в несколько туров. На первом туре каждый эксперт дает числовую оценку объекту. После этого исследователь подсчитывает и сообщает всем экспертам среднюю оценку (медиану) и показатель разброса оценок (ин-

тервал между крайними квартилями). Экспертов, давших крайние оценки, просят письменно обосновать свое мнение. С этими обоснованиями исследователь знакомит (анонимно) остальных экспертов, после чего аналогичным образом проводится второй тур опроса. Подобные итерации заканчиваются тогда, когда, по мнению исследователя, будет достигнуто «достаточное» согласие между оценками экспертов. Помимо тривиального правила остановки (при полном совпадении всех оценок), обычно не выполняемого, литература по процедуре Делфи не содержит четких формальных правил остановки.

Этот исходный вариант процедуры, называемый иногда стандартным Делфи, повлек за собой множество разновидностей и модификаций. Большая часть модификаций обусловлена конкретной областью приложения метода Делфи и выразилась в специальной постановке вопросов. Так, в процедуре, ориентированной на политические проблемы Policy Delphi, акцент делается на формулировке проблемы и выявлении и оценке вариантов ее решения. В применении метода Делфи к задаче прогнозирования производства стали [42] от экспертов запрашивались графики выпуска стали на предстоящие годы. Другие содержательные задачи с соответствующими формулировками вопросов описаны в [43—45] и многих других публикациях. Подчеркнем, что метод Делфи следует понимать прежде всего как способ организации опроса, не связанный с видом экспертных суждений.

Другая часть работ охватывает вопросы технической реализации процедуры Делфи. Здесь, в первую очередь, выделим работы, в которых предлагается взаимодействие экспертов посредством ЭВМ [46—48].

Последняя группа модификаций направлена на усовершенствование самой процедуры Делфи. Имеются в виду работы, не переосмысливающие исходные положения этой процедуры, а содержащие частные ее видоизменения. Из них укажем на «упрощенный Делфи» [49], представляющий компромисс между методом Делфи и дискуссией. Эксперты общаются за круглым столом, но устно происходит лишь выступление экспертов, давших крайние оценки. Остальное — отделенные друг от друга туры, письменные ответы, обратная связь — повторяет стандартный метод Делфи. Кроме желания провести опрос побыстрее и попроще, этому способу трудно дать иную мотивировку.

Другие исследователи исходят из того, что экспертам сложно давать точную количественную оценку объекта, а проще указать интервал, в котором лежит оценка. В методе Делфи-П [50] вся процедура остается той же, но эксперт дает каждый раз высшую, среднюю и низшую оценку; соответственно по обратной связи передаются медианы этих трех оценок.

В методе Форда [51] предлагается более изощренная схема, направленная на уменьшение следования экспертов сред-

ней групповой оценке. На первом туре эксперт сообщает верхнюю и нижнюю границы, в которых лежит точная, по его мнению, оценка. Подсчитываются средние высшая и низшая границы и середина полученного интервала. На втором туре эта средняя точка сообщается экспертам, которых просят указать, выше или ниже ее лежит точная оценка. По моде ответов устанавливается одна из двух областей, например верхняя. Она теперь ограничена сверху предыдущей верхней границей и снизу предыдущей средней точкой. На третьем туре середина этого нового интервала адресуется экспертам с аналогичной просьбой и т. д. Вопрос, по идее, должен быть остановлен при достижении очень малого интервала.

Работы этого плана не получили дальнейшего развития, возможно, потому, что каждая новая прикладная задача может потребовать совсем другого вида ответа, отличного, скажем, от точечной числовой оценки. Кроме того, правомерность отказа от такой оценки не получила серьезного обоснования. Более важными оказались вопросы, связанные с основами организации работы коллектива экспертов. Как раз здесь появились методы, отходящие от исходных положений метода Делфи.

Этот отход связан в основном с критическим осознанием того факта, что оценки экспертов в методе Делфи обычно сближаются. Причина этого — по существу заложенное в методе давление на экспертов с целью приближения их оценок к среднегрупповой. Ведь каждому эксперту неоднократно сообщают среднюю оценку; очевидно, что у эксперта нет оснований отклоняться от нее дальше, зато есть основания приблизиться к ней. Эти основания зафиксированы психологами в виде явления «следование мнению большинства». Таким образом, мы сталкиваемся с тем же групповым мышлением, которое, хотя и в ослабленной форме, проявляется и при заочном общении членов группы [4].

В [52] с целью сближения оценок экспертов предлагается метод парных взаимодействий. Априори эксперты дают исходные оценки, после чего начинаются итерации. На каждой из них происходит случайное разбиение всех экспертов на пары, в каждой паре эксперты обмениваются оценками. Один (любой) эксперт в паре пересматривает оценку; когда это произойдет во всех парах, осуществляется переход к следующей итерации. Снова случайно образуются новые пары и т. д. Правило остановки в [52] не определено, но, видимо, процедура завершается тогда, когда все пары выдают совпадающие между собой или близкие, по мнению исследователя, оценки. В [52] описаны эксперименты на ЭВМ в рамках следующей модели. Считается, что с вероятностью $(1-\varepsilon)$ эксперт меняет оценку так, что уменьшает расхождение ρ с соседом по паре, и с вероятностью ε не меняет оценки; эксперту в пределах пары разрешается делать не более m шагов навстречу соседу. Если, на-

пример, эксперты ранжируют объекты, то ρ — общее число инверсий от перестановки одного эксперта до перестановки другого, а m — число транспозиций в перестановке эксперта. Параметр ε характеризует «упрямство» экспертов, а параметр m — «степень соглашательства». Данная работа возникла из проблематики взаимодействия автоматов. В контексте же рассматриваемой проблематики можно считать, что основным отличием этой процедуры является локальное взаимодействие экспертов.

В [53] предложена так называемая последовательная процедура опроса. Она осуществляется следующим образом: на первом туре случайно выбирается один из имеющейся совокупности экспертов; он дает оценку объекта и отдельно от нее текст, содержащий известные эксперту сведения и мнения о различных особенностях оцениваемого объекта, которые эксперт принимал во внимание при вынесении оценки. На втором туре выбирается случайным образом второй эксперт, который также дает оценку и соответствующий текст. Первый и второй эксперты знакомятся с текстами (но не оценками) друг друга, после чего сообщают исследователю свои скорректированные под воздействием этих текстов оценки. На третьем туре появляется и дает оценку и ее текстовое обоснование третий эксперт, он получает тексты первого и второго экспертов, а те — текст третьего эксперта, затем все трое сообщают свои новые оценки и т. д. Так продолжается до того момента, когда оценки введенных экспертов перестанут изменяться. Например, процедура может быть остановлена тогда, когда k уже выбранных экспертов на протяжении l туров не меняли своих оценок, несмотря на получение новых текстов. Существенно, что каждый эксперт только один раз (в момент ввода в процедуру) дает текст (обосновывающий начальную оценку), в дальнейшем эксперт только получает подобные тексты других экспертов и корректирует оценку.

В отличие от описанных выше данная процедура нацелена не на сближение оценок экспертов, а на стабилизацию экспертами своих индивидуальных оценок, которые могут и существенно различаться между собой. Результатом использования последовательной процедуры является набор итоговых суждений, возможно, различных между собой, но таких, что на них уже не влияет (или мало влияет) информация других экспертов. Из данного подхода вытекает принципиально иное правило остановки процедуры, которое может быть легко формализовано. Особенность этой процедуры заключается также в том, что каждый эксперт получает по обратной связи не оценку, а обоснования оценок других экспертов. Наконец, надо отметить, что взаимодействие экспертов здесь в основном локальное: на каждом туре все введенные эксперты получают информацию только одного (вводимого) эксперта.

В [54] указано, что присущая методу Делфи количественная

обратная связь, т. е. сообщение экспертам количественных результатов предыдущего тура, приводит к искусственному сближению оценок экспертов и искажает их действительные точки зрения. Для устранения этого предлагается процедура «качественной обратной связи», состоящая в следующем. На первом туре все эксперты одновременно дают оценки и подробные текстовые обоснования своих оценок. Затем составляется и передается каждому эксперту сводная информация, суммирующая обоснования всех экспертов (это и называется «качественной обратной связью»). На втором туре снова каждый эксперт дает и оценку, и ее текстовое обоснование. Такие итерации повторяются до тех пор, пока индивидуальные оценки экспертов не стабилизируются.

Как видно, эта процедура сходна с последовательной, но есть отличия. Во-первых, если в последовательной процедуре каждому эксперту передается каждый раз информация одного эксперта, то здесь — сводная суммарная информация, которая вряд ли является хорошим стимулом для пересмотра оценки. Во-вторых, в отличие от последовательной процедуры в данном подходе эксперт на каждой итерации заново дает не только оценку, но и обоснование. Последнее усложняет работу экспертов и, возможно, необязательно, т. к. на первом туре эксперт обосновывает свою личную оценку, а последующие изменения оценки (и ее обоснования) обусловлены текстовой информацией, полученной от других экспертов.

Процедура индивидуальной обратной связи [55] доводит до логического завершения идею локализации обратной связи. На первом туре эксперты дают оценки и их текстовое обоснование. На втором в качестве датчика выбирается какой-то один эксперт, и его мнение, оценки и обоснования, передаются по обратной связи всем остальным экспертам. Последние анализируют полученное мнение и сообщают свои скорректированные оценки (без обоснований). Аналогично содержание последующих туров, на каждом из которых по обратной связи передается мнение одного (нового) эксперта. При отсутствии данных об информированности экспертов в рассматриваемой проблеме выбор датчиков производится случайным образом. В противном случае вначале выбираются более информированные эксперты. Процедура останавливается при наступлении стабилизации суждений.

Достоинство индивидуальной обратной связи состоит в том, что она уменьшает возможность механической подстройки под групповое мнение. Индивидуальное мнение привычнее и легче воспринимать и перерабатывать. Оно передается без преобразований, что неизбежно, скажем, в методе Делфи. Эту схему гораздо проще реализовать, т. к. здесь отсутствует этап обработки мнений. К недостаткам процедуры можно отнести возможно

большее, чем при методе Делфи, число туров и влияние на результаты опроса порядка назначения датчиков.

Заканчивая обзор методов коллективного получения знаний, зададимся вопросом о сравнительных достоинствах и недостатках рассмотренных методов. Ответ на этот вопрос дается в работе [56]. Здесь обобщены экспериментальные исследования двух групп: 1) посвященные сравнению двух классов методов — с непосредственным взаимодействием и без него, 2) посвященные сравнительному изучению методов второго класса. В экспериментальных исследованиях, о которых идет речь, экспертам, участвующим в эксперименте, предъявляется серия тестовых вопросов, ответы на которые исследователь заранее знает. Основным критерием качества процедур служит точность оценок экспертов — их близость к истинным значениям. Кроме того, учитываются степень субъективной уверенности экспертов в своих ответах, затраты времени на реализацию процедуры, трудоемкость ее проведения для экспертов и исследователя, оценка самими экспертами качества процедур.

Приведем только выводы из сравнительного анализа. Сначала дадим выводы о сравнении двух классов методов.

1. Методы косвенного взаимодействия (имеется в виду главным образом метод Делфи) показывают в целом несколько большую точность ответов экспертов, чем методы с прямым взаимодействием (главным образом «дискуссия за круглым столом»).

2. В методах типа Делфи не было эффекта явного повышения точности от тура к туру.

3. Разброс оценок экспертов при методе Делфи от тура к туру существенно уменьшается.

4. Субъективные отзывы экспертов показывают предпочтительнее ими традиционного общения лицом к лицу. Это можно объяснить привычкой такого стиля тесного общения, дающего эксперту чувство удовлетворенности от зримой групповой работы.

Теперь приведем выводы, связанные с изучением только методов второго класса. Сюда относятся как прямые сравнения методов между собой, так и исследования отдельных вопросов, связанных с их реализацией.

1. В целом итерации в описанных методах обеспечивают заметное повышение точности ответов, причем между существующими методами в этом отношении нет значительного отличия. Однако большую точность дает обратная связь, предусматривающая передачу и экспертных оценок, и их обоснований.

2. Эффект сближения оценок экспертов более всего проявляется в методе Делфи под действием групповой количественной обратной связи. Но достижение согласия в этом случае не гарантирует точности ответов.

3. Достижение значительного согласия индивидуальных оце-

нок не является надежным правилом останковки итераций. Более надежным правилом представляется останковка при стабилизации оценок каждого эксперта.

4. Субъективные ощущения экспертов показывают предпочтение процедуры Делфи, что подтверждает: большая включенность эксперта в работу всей группы порождает у него удовлетворенность процедурой, которая на деле может уводить от точного ответа.

5. По трудоемкости и затратам времени лучшей оказалась процедура индивидуальной обратной связи.

В целом в [56] выносится заключение о превосходстве методов косвенного взаимодействия над методами с непосредственным взаимодействием. Там же вычлняются основные проблемы, связанные с дальнейшим совершенствованием методов косвенного взаимодействия.

Коллективные методы достаточно легко поддаются автоматизации. В особенности это справедливо для коллективных методов второго класса, легко допускающих работу исследователя за центральным, а экспертов за периферийными терминалами ЭВМ. Кроме уже упомянутых в этой связи работ [46—48], можно назвать и работу [57], где намечаются контуры распределенной системы реального времени, в которой датчики входной информации — эксперты работают за терминалами ЭВМ. Исследователь ведет работу (диалог) с ЭВМ, в ходе которого эксперты время от времени получают конкретные вопросы и отвечают на них. При этом ЭВМ предлагает различные методы проведения опроса, ведет обработку поступающих ответов, а исследователь выбирает методы и принимает решения о ходе опроса, в том числе решение о его останковке.

5. ФОРМЫ ПОЛУЧАЕМЫХ ЗНАНИЙ

Выше рассматривались методы, не ориентированные на какие-то конкретные формы знаний. Теперь мы затронем вопрос о формах, в которых извлекаются знания. Сразу оговоримся, что по-прежнему разделяем задачи получения и представления знаний и будем говорить не о формах представления, а о формах получаемых знаний. Этот вопрос, безусловно, связан с методами получения знаний, но в литературе обсуждается как самостоятельный. Надо отметить, что во всей рассматриваемой в данном обзоре проблематике данный вопрос, пожалуй, наименее исследован. В частности, мало изучена связь форм знаний с методами их получения.

Конечно, как и раньше, мы не будем касаться конкретного содержания знаний, определяемого решаемой задачей. Мы будем говорить о достаточно общих формах, в которых знания

извлекаются у экспертов. Далее на основе этих знаний решается задача представления знаний.

Можно выделить три преобладающие точки зрения по обсуждаемому вопросу и соответственно три направления исследований.

Первая состоит в том, что эксперт может сам связно и понятно сформулировать свои знания, поэтому он не нуждается в каких-то формах, предлагаемых исследователем. Процедура состоит в том, что эксперты (посредством одного из описанных выше методов) самостоятельно сообщают свои знания, после чего исследователь анализирует их и трансформирует в окончательную форму представления. Преимущество такого подхода в том, что экспертные знания не требуется укладывать в прокрустово ложе готовых форм, которые могут ограничивать или даже сводить на нет ценность получаемой информации. В [31] отмечено, что экспертам не свойственно выражать знания в специальных формах, обсуждаемых в литературе; эти формы лишь могут быть построены из таких содержащихся в естественной речи экспертов категорий, как факты, цели, условия и т. п.

В первом подходе обходится первая из указанных выше трудностей получения знаний — расхождение между присущими эксперту способами выражения знаний и требуемыми формами представления. Недостаток этого подхода в том, что эксперт не всегда может ясно выразить свои знания и они будут потеряны для исследователя. Иначе говоря, препятствием является вторая трудность получения знаний — неспособность к вербализации.

Вторая точка зрения состоит в том, что не следует полагаться на способности эксперта, а надо ориентироваться на заранее выбранные формы представления знаний. По существу экспертов принуждают сообщать знания сразу в той форме, которая будет использоваться при представлении. По сравнению с первым подходом преимущества и недостатки здесь меняются местами: плюс в том, что обходятся трудности вербализации, а минус в использовании ограничительных для эксперта форм выражения знаний.

На практике данный подход используется довольно активно. Например, при разработке экспертных систем эксперты (при помощи инженера по знаниям) формулируют знания в виде правил продукции или фреймов, т. е. в окончательных формах представления. При проведении экспертных сцепок очень часто от экспертов требуют информацию также в итоговых формах: ранжирования, классификации и т. п. Однако в последнее время эта точка зрения подвергается серьезной критике, особенно в литературе по экспертным системам. Фактически форма знаний навязывается эксперту, при этом игнорируется качество получаемой информации. Иначе говоря, игнорируется первая трудность получения знаний. Например, в [31] прямо говорится,

что экспертам не свойственно выражать знания в таких формах, как правила продукции или фреймы.

Третья точка зрения базируется на допущении, что знания экспертов имеют определенные формы внутренней организации, на которые и надо опираться при извлечении знаний; эти формы следует также учитывать и при представлении знаний. Эксперты сообщают знания в виде ответов на специальные вопросы, т. е. в специальной форме, после чего исследователь преобразует знания в окончательную форму представления. Здесь преимущество в том, что обходятся трудности вербализации. Что же касается первой трудности, то сторонники этой точки зрения полагают, что они используют не ограничительные, а присущие экспертному мышлению формы выражения знаний. Данная точка зрения в настоящее время представляется убедительной и изучается более активно, чем две другие. Поэтому мы посвятим ей оставшуюся часть данного раздела.

Итак, сторонники этого направления не полагаются на способность экспертов к непосредственному выражению знаний. Экспертов просят дать суждения по определенной форме, из которой исследователь строит окончательную форму представления знаний. Методы этого направления опираются на различные допущения о формах организации человеческих знаний. Эти допущения должны быть обоснованы и проверены прежде всего в психологическом плане, излагаемые ниже методы такую проверку прошли.

Преимущества методов этого направления обосновываются в [14]. Эти методы тесно связаны с когнитивной психологией, давно изучающей проблемы представления и передачи человеческого опыта. Известны работы по изучению принципов хранения и переработки информации человеком. В последнее время разработаны алгоритмы психологического шкалирования, позволяющие эмпирическим путем порождать различные формы представления знаний на основе данных, полученных от человека. Наряду с этим большое внимание уделяется анализу механизмов человеческого опыта. В частности, оценивались различия между «новичками» и опытными специалистами в способности к распознаванию образов, организации памяти, решению проблем. Особо важны психологические исследования, в которых формальные методы представления знаний с помощью алгоритмов шкалирования сочетаются с человеческим опытом. Цель этих исследований — в применении различных алгоритмов психологического шкалирования для порождения представлений в хорошо определенных областях знаний, причем для лиц с различным профессиональным опытом.

Методы познания механизмов человеческого опыта могут использоваться и для получения знаний. Ряд исследователей считает, что метод интервьюирования не подходит для выявления некоторых видов знаний. Вообще говоря, различным видам зна-

ний в идеале должны соответствовать разные методы и формы их получения. Поэтому в рамках когнитивной психологии должен быть создан инструментарий, включающий в себя методы для выявления знаний разных видов и их представления. Важно, что каждый метод следует использовать только для строго определенных областей приложения [14]. Попытка создания такого инструментария и делается в обсуждаемых методах.

Многомерное шкалирование. Этот метод, подробно описанный в [58], исходит из допущения, что знания представляются в геометрическом многомерном пространстве. Понятно, что этот подход должен применяться только к знаниям, допускающим такое представление. Задача эксперта — дать суждения о степени сходства (близости) между всеми парами объектов (понятий) в рассматриваемой области. Обычно суждения выдаются с помощью специальных шкал близости.

При анализе суждений используется гипотеза о том, что различия между объектами объясняются расхождением по небольшому числу признаков, и человек, вынося свои суждения, неявно учитывает эти признаки. Задача состоит в том, чтобы на основе анализа суждений выявить признаки, которыми руководствовался человек, и определить, какими значениями этих признаков характеризуется каждый объект. Для этого используется геометрическое представление объектов в виде точек в пространстве небольшой размерности. Нужно построить такое психологическое пространство и разместить в нем точки-объекты таким образом, чтобы расстояния между ними наилучшим образом в смысле некоторого критерия соответствовали полученным различиям.

Ответы эксперта обычно записываются в квадратной матрице, по строкам и столбцам которой стоят объекты, а элементы показывают оценки близости. Данные матрицы обрабатываются с помощью машинных программ для получения искомого геометрического представления. Последнее может быть предъявлено эксперту на предмет дополнительного визуального анализа: выявления классов близких объектов, далеко отстоящих объектов и т. п.

Основная трудность применения этого метода — быстрый рост числа сравнений (пар) по мере роста числа объектов; при довольно большом числе объектов загрузка эксперта становится непомерно великой. Одновременно существенно усложняется и поиск геометрического представления [26].

В [59] делается попытка связать многомерное шкалирование с механизмами представления знаний человеком. Особая роль при этом отводится памяти. Понятия, оставляющие какую-то область, хранятся в памяти в определенной системе. Согласно одной из моделей памяти, теоретико-множественной, все имеющиеся понятия группируются на подмножества близких понятий. Понятия характеризуются набором признаков. Сравнимая

понятия, человек сопоставляет их значения по признакам. Объяснение того, почему многомерное шкалирование дает содержательно интерпретируемые результаты, следует искать в психофизиологии. А именно, тот факт, что в результате анализа данных о субъективных сходствах получают систему независимых признаков, согласуется с выводами психофизиологии о кодировании сигнала в нервной системе вектором возбуждения независимых нейронных механизмов.

Кластер-анализ. При этом методе от эксперта также просят матрицу попарных близостей между объектами, однако в основу положены иные допущения. Предполагается, что объекты образуют иерархическую систему классов и каждый объект принадлежит к одному из них. Имеется немало алгоритмов построения иерархической классификации по матрице близостей. Например, согласно алгоритму [60], вначале два самых близких объекта объединяются в один класс, рассматриваемый как один объект. В соответствии с этим строится новая матрица близостей; в ней снова ищутся два ближайших объекта и т. д.

Достоинство иерархического кластер-анализа в том, что он может быть осуществлен и простым ручным способом. Недостаток тот же, что в предыдущем методе — трудоемкость работы эксперта при достаточно большом числе объектов. Кроме того, выбор алгоритма кластеризации довольно произволен; между тем разные алгоритмы могут приводить к резко различным иерархиям. В этом смысле результаты анализа могут быть в известной степени субъективными.

Сетевая структура. Входными данными для нее также могут быть полученные от эксперта суждения о попарных близостях. Допускается, что эти данные отражают внутреннее представление эксперта об объектах как о сети взаимосвязанных элементов. Вершины сети представляют объекты (явления, понятия), а дуги — отношения между ними. Это представление по-разному конкретизируется в сетях различного вида. Например, используется сеть, в которой между каждыми двумя вершинами есть одна основная дуга, и, кроме того, для некоторых пар вершин есть вторичные дуги.

Возможны различные подходы к построению сети из данных о близостях. Например, так называемая минимально связанная сеть строится путем соединения дугами только максимально близких между собой объектов [26].

В некоторых исследованиях показано, что эксперты формируют довольно ясные и простые сети, выгодно отличающиеся в этом от неспециалистов. По-видимому, этот метод адекватно отражает некоторые существенные аспекты внутренней организации знаний у экспертов [26].

Деревья из памяти (trees from recall). Этот метод изложен в [61], посвященной изучению различий в организации памяти у экспертов и неопытных лиц. В отличие от выше при-

веденных методов, от эксперта не требуются суждения о сходстве объектов. Как и в кластер-анализе, здесь допускается, что объекты образуют классы. Но этот метод идет дальше и допустит допущение о том, как человек формирует представление о классах. Именно, предполагается, что человек, формируя такое представление в какой-то области, обращается к своей памяти: сначала он вспоминает объекты одного какого-то класса, затем другого и т. д.

В этом методе от эксперта требуют просто вспомнить и назвать по порядку объекты данной области. Поскольку названный порядок может быть случайным, эксперта просят сделать это многократно. Все такие попытки фиксируются и совместно анализируются на предмет выявления закономерности. Объекты, которые устойчиво вспоминаются вместе (один за другим), включаются в один класс. При этом случается, что есть отдельные классы, внутри которых объекты называются в определенном порядке. В других классах порядок названия варьируется.

Результат такого анализа может быть графически показан деревом, на нижнем уровне которого находятся все названные объекты, объединенные в классы, как сказано выше. Отдельные классы (с внутренним порядком) можно пометить. Дальнейшая иерархия (объединение соседних классов) строится так: объединяется та пара классов, которые чаще вспоминались один за другим, и т. д.

В экспериментах [61, 62] показано, что упомянутые деревья существенно отличались для экспертов и неспециалистов. В частности, между деревьями различных экспертов оказалось гораздо больше сходства, чем между деревьями «новичков».

Устойчивая решетка (repertory grid). Метод возник в рамках теории персональных конструкторов в области клинической психологии [63]. Там он служил выявлению признаков, которые характеризуют эмоциональное состояние пациентов. В [64, 65] метод переориентирован на приобретение знаний в экспертных системах.

Предполагается начальное интервью с экспертом, в ходе которого он называет некоторые объекты в рассматриваемой области. Исследователь берет три из них и просит эксперта указать, какие признаки отличают любые два объекта от третьего. Остановившись на каких-то признаках, эксперт сообщает далее их значения (градации). Затем исследователь берет другую тройку объектов и просит сделать для них то же самое. Тройки сменяют одна другую до тех пор, пока исследователь не убедится, что выявлены все важные признаки. При этом значения признаков подбираются так, чтобы они составили «биполярные» шкалы. Такие шкалы имеют противоположные по смыслу крайние значения, например уверенный — неуверенный, активный — пассивный, организованный — неорганизованный.

Далее исследователь строит «решетку», т. е. таблицу, по

столбцам которой записываются объекты, а по строкам признаки. Эксперта просят дать каждому объекту оценку по каждому из признаков (в биполярных шкалах).

Анализ полученной решетки включает в себя классификацию объектов и классификацию признаков. Обе классификации могут быть иерархическими. В [64, 65, 66] решетки используются для построения правил продукций, включаемых в базу знаний экспертной системы. Решетки могут быть и средством обсуждения разногласий между разными экспертами. Сравнение различных решеток может помочь экспертам понять причины расхождений и устранить их.

Из сказанного ясно, что этот метод исходит из допущения о хранении человеческих знаний в виде описанной решетки. Поэтому, заканчивая обзор этой группы, стоит еще раз подчеркнуть, что они основаны на допущениях об организации знаний у человека. Полезность этих методов проявится в той степени, в какой будут справедливы эти допущения для данного вида знаний.

Следует сказать, что в статье [26] предложено деление методов получения знаний на два класса: прямые и не прямые. К прямым отнесены интервьюирование, анкетирование и т. д., а к непрямым — многомерное шкалирование и другие методы этого направления. На наш взгляд, здесь смешаны методы и формы получения знаний. Интервьюирование и т. д. — это методы получения, а многомерное шкалирование и т. д. — это формы организации знаний. Для получения последних, судя по литературе, используются в основном такие методы, как анкетирование и интервьюирование.

Сказанное показывает важность проработки такого мало изученного вопроса, как связь методов и форм получения знаний. Не пытаясь сколько-нибудь полно его осветить, сделаем только замечание относительно индивидуальных методов получения знаний. Методы интервьюирования и анкетирования можно отнести к универсальным — их можно использовать для получения всех перечисленных в этом разделе форм знаний. Методы же наблюдений за экспертом, анализа протоколов и анализа прерываний ориентированы на свободную форму, когда эксперт самостоятельно формулирует знания.

ЗАКЛЮЧЕНИЕ

1 Проблема получения экспертных знаний становится одной из важнейших проблем в области управления. Актуальность этой проблемы прежде всего объясняется широким распространением экспертных систем. Однако названная проблема имеет гораздо более долгую историю, связанную с различными видами человеческой деятельности. В настоящее время двумя основными областями использования методов получения знаний

являются экспертные системы и организация реальных (очных) экспертиз.

2. В процессе приобретения знаний следует выделить два этапа: получение (извлечение) знаний и их представление. Первый ориентирован непосредственно на работу с источником знаний — экспертами. Второй обусловлен специфической формой представления знаний в системе. Противоречие между первым и вторым этапами составляет важную особенность инженерии знаний и предопределяет значение посредника между экспертами и системой.

3. Независимо от области использования получение знаний — специфический процесс, имеющий свои внутренние особенности, закономерности и трудности. К основным трудностям относятся расхождение между способами выражения знаний экспертами и формой представления; ограниченная способность эксперта к вербализации знаний; систематические смещения (противоречия, непоследовательности), проявляемые экспертом при сообщении знаний; неточность, неполнота и другие несовершенства экспертных суждений.

4. Исследования проблемы получения знаний, в основном проводимые в когнитивной психологии и социальной психологии, позволяют формировать инструментарий методов получения знаний, которые могут использоваться в разных областях и задачах с учетом их специфики. При этом для определения области применимости нужна классификация методов получения знаний.

5. Полезно выделить методы индивидуального и коллективного получения знаний. В первом случае источником знаний является один эксперт, во втором — экспертная группа. Методы индивидуального получения знаний применяются главным образом в экспертных системах, а коллективного — при организации реальных экспертиз.

6. К индивидуальным методам относятся интервьюирование, анкетирование, анализ протоколов и другие методы, предполагающие совместную работу эксперта и исследователя (посредника) и отличающиеся характером этой работы и соотношением ролей участников.

7. Коллективные методы можно разделить на методы, допускающие и исключающие прямое взаимодействие членов экспертной группы. К первому классу методов относятся такие традиционные способы прямого общения, как дискуссия за круглым столом. В методах второго класса (метод Делфи и его модификации) эксперты изолированы друг от друга, опрос проводится в несколько туров, на каждом из которых эксперты посредством исследователя получают информацию о позициях всех участников опроса.

8. Помимо методов получения знаний необходимо исследование форм, в которых знания извлекаются у экспертов. Это

могут быть свободные формы, опирающиеся на способность экспертов самостоятельно излагать знания, или заранее выбранные формы, например базирующиеся на гипотезах о внутренней организации знаний в памяти человека.

9. Развитие инструментария получения знаний требует выработки более совершенной классификации методов получения и форм извлекаемых знаний, которая позволит выбирать методы и формы, согласованные между собой и соответствующие решаемой задаче.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ ИНФОРМАЦИИ

1. Бешелев С. Д., Гурвич Ф. Г. Экспертные оценки.— М.: Наука, 1973.— 160 с.
2. Евланов Л. Г., Кугузов В. А. Экспертные оценки в управлении.— М.: Экономика 1978.— 133 с.
3. Литвак Б. Г. Экспертная информация: методы получения и анализа.— М.: Радио и связь, 1982.— 184 с.
4. Панкова Л. А., Петровский А. М., Шнейдерман М. В. Организация экспертизы и анализ экспертной информации.— М.: Наука, 1984.— 120 с.
5. Уинстон П. Искусственный интеллект.— М.: Мир, 1989.— 519 с.
6. Нильсон Н. Принципы искусственного интеллекта.— М.: Радио и связь, 1985.— 376 с.
7. Построение экспертных систем / Под ред. Хейес-Рота Ф., Уотермана Д., Лената Д.— М.: Мир, 1987.— 441 с.
8. Попов Э. В. Экспертные системы: решение неформализованных задач в диалоге с ЭВМ.— М.: Наука, 1987.— 288 с.
9. Уотерман Д. Руководство по экспертным системам.— М.: Мир, 1989.— 388 с.
10. Стошнер Т. Индустрия знаний // Экспертные системы. Принципы работы и примеры / Под ред. Р. Форсайта.— М.: Радио и связь, 1987.— С. 197—212.
11. Gaines B. R. An overview of knowledge-acquisition and transfer // Int. J. Man-Mach. Stud.— 1987.— 26.— P. 453—472.
12. Nisbett R. E., Wilson T. D. Telling more than we can know: verbal reports on mental processes // Psychological Rev.— 1977.— 84.— P. 231—259.
13. Hawkins D. An analysis of expert thinking // Int. J. Man-Mach. Stud.— 1983.— 18.— P. 1—47.
14. Кук Н. М., Макдоналд Д. Э. Формальная методология приобретения и представления экспертных знаний // ТИЭР.— 1986.— № 10.— С. 145—155.
15. Bainbridge L. Verbal reports as evidence of the process operator's knowledge // Int. J. Man-Mach. Stud.— 1979.— 11.— P. 411—436.
16. Hankins G. Expert systems and human learning // Int. J. Appl. Eng. Educ.— 1988.— № 6.— P. 511—514.
17. Ларичев О. И. Объективные модели и субъективные решения.— М.: Наука, 1987.— 143 с.
18. Jacob V. S., Gaultney L. D., Salvendy G. Strategies and biases in human and decision making and their implications for expert systems // Behavior and Inf. Technol.— 1986.— 5.— P. 119—140.
19. Evans J. Human biases and computer decision-making: a discussion of Jacob et al. // Behavior and Inf. Technol.— 1987.— 6.— P. 483—487.
20. Lehner P. E., Adelman L. Biases in knowledge engineering // Proc. IEEE Int. Conf. on Syst., Man and Cybern. Alexandria, 1987.— N. Y., 1987.— 1.— P. 376—377.
21. Gammack J. G., Young R. U. Psychological techniques for eliciting expert knowledge // Research and Development in Expert Syst.— Cambridge: Univ. Press, 1984.— P. 105—112.

22. Grover M. D. A pragmatic knowledge acquisition methodology // The Eight Int. Joint Conf. on Artificial Intelligence.— Karlsruhe, 1983.— P. 436—438.
23. Тюрин Ю. Н. Непараметрические методы статистики.— М.: Знание, 1978.— 73 с.
24. Раушенбах Г. В., Филиппов О. В. Экспертные оценки в медицине (обзор).— М.: ВНИИ медицинской и медико-технической информации, 1983.— 80 с.
25. Шмерлинг Д. С., Дубровский С. А., Аржанова Т. Д., Френкель А. А. Экспертные оценки. Методы и применение (обзор) // Статистические методы анализа экспертных оценок.— М.: Наука.— 1977.— С. 290—382.
26. Olson J. R., Rueter H. H. Extracting expertise from experts: methods for knowledge acquisition // Expert Syst.— 1987.— 4.— P. 152—168.
27. Davies M., Hakiel S. Knowledge harvesting: a practical guide to interviewing // Ibid.— 1988.— 5.— P. 42—50.
28. Бешелев С. Д., Гурвич Ф. Г. Математико-статистические методы экспертных оценок.— М.: Статистика, 1974.— 159 с.
29. Ericsson K. A., Simon H. A. Protocol analysis: verbal reports as data.— Cambridge: MIT Press, 1984.— 170 p.
30. Newell A., Simon H. A human problem solving-Englewood cliffs.— New Jersey: Prentice-Hall, 1972.— 290 p.
31. Anjewierden A. Knowledge acquisition tools // AI communications.— 1987.— № 1.— P. 29—38.
32. Dressler F. Subjective methodology in forecasting // Technol. Forecasting and Social Change.— 1972.— 3.— P. 427—439.
33. Козелецкий Ю. Психологическая теория решений.— М.: Прогресс, 1979.— 504 с.
34. Janis I. L. Victims of groupthink: a psychological study of foreign policy decisions and fiascoes.— Boston: Houghton Mifflin, 1972.— 253 p.
35. Янч Э. Прогнозирование научно-технического прогресса.— М.: Прогресс, 1970.— 568 с.
36. Clark C. H. Brainstorming—the dynamic new way to create successful ideas.— Garden City: Double day, 1958.— 170 p.
37. Йорданский Д. И. Метод группового обсуждения с отнесением оценки // Всесоюзное совещание по количественным методам в социологии.— М.: ИКСИ АН СССР, 1967.— С. 49—59.
38. Van de Ven A., Delbeeg A. Nominal versus interacting group processes for committee decision—Making Effectiveness // Academy of Manag. J.— 1971.— 14.— P. 203—212.
39. Dalkey N., Helmer O. Experimental application of the delphi method to the use of experts // Manag. Sci.— 1963.— 9, № 3.— P. 458—467.
40. The Delphi method: Techniques and applications.— Reading, Mass., Addison-Wesley, 1975.— 620 p.
41. Turoff M. The policy Delphi // Ibid.— P. 84—101.
42. Goldstein N. H. A Delphi on the future of the steel and ferroalloy industries // Ibid.— P. 210—226.
43. Milkovich G., Annoni A., Makoney T. The use of the Delphi procedures in manpower forecasting // Manag. Sci.— 1972.— 19, № 4.— P. 381—388.
44. Middendorf W. A. Modified Delphi method of solving bassiness problems // IEEE Trans. Eng. Manag.— 1973.— EM-20, № 4.— P. 130—133.
45. Jillson J. A. The national drug—abuse policy Delphi: Progress report and findings to date // The Delphi method: Techniques and applications. Reading Mass.: Addison-Wesley, 1975.— P. 124—159.
46. Turoff M. Delphi conferencing: Computer based conferencing with anonymity // Technol. Forecasting and Social Change.— 1972.— 3, № 2.— P. 159—204.
47. Price C. R. Conferencing via computer: Cost effective communication for the era of forced choice // The Delphi method: Techniques and applications.— Reading, Mass.; Addison-Wesley, 1975.— P. 497—516.
48. Давыденко А. А. Программное обеспечение проведения экспертного опроса с регулируемой обратной связью // II Всесоюзная конф. по статисти-

ЭВРИСТИЧЕСКИЕ МЕТОДЫ В КАЛЕНДАРНОМ ПЛАНИРОВАНИИ

А. Д. Плитман, М. И. Рубинштейн

ВВЕДЕНИЕ

В организации процессов различного типа (производственных, транспортных, вычислительных и др.) роль планирования сводится к разработке программы взаимодействия во времени и пространстве различных элементов процесса. Задачи планирования, состоящие в разработке программ, в которых ведущую роль играет временной фактор, или, более точно, организация взаимодействия элементов рассматриваемого процесса во времени, называют задачами календарного планирования.

Календарное планирование играет важную роль в управлении современным производством и обеспечении его эффективности. В будущем роль его еще более возрастет в связи со все большим распространением высокопроизводительного оборудования, во многих случаях включаемого в состав автоматизированных производственных систем.

Практика календарного планирования при использовании формализованных методов, как правило, приводит к постановке и необходимости решения задач оптимизации комбинаторного характера. Эти задачи в большинстве случаев являются *NP*-полными. Другая особенность задач календарного планирования, которая проявляется при их формализации, состоит в «огрублении» исходных данных в связи с необходимостью дискретизации времени. Наконец, еще одна особенность задач календарного планирования связана с трудностями построения компактных формальных моделей простой структуры. Все эти особенности задач календарного планирования обуславливают особую роль эвристических методов их решения.

В настоящем обзоре рассматриваются следующие основные направления применения эвристических методов в календарном планировании. В рамках первого направления описываются эвристические методы, основанные на анализе формальных моделей реальных задач календарного планирования, т. е. данное направление затрагивает класс задач, для которых могут быть построены обозримые, поддающиеся анализу модели. Здесь анализируются основные эвристические приемы построения приближенных алгоритмов решения задач комбинаторного программирования и их реализации для задач календарного планирования.

Второе направление составляют эвристические процедуры планирования, опирающиеся на использование имитационных

ческому и дискретному анализу нечисловой информации и экспертным оценкам.— М.: ВНИИСИ. 1984.— С. 359.

49. Гордон Т. Дж. Новые подходы к методу «Делфи» // Научно-техническое прогнозирование для промышленности и правительственных учреждений.— М.: Прогресс, 1972.— С. 84—99.
50. Brown B., Cochran S., Dalkey N. The Delphi method II: Structure of experiments // Memorandum RM-5957—PR.— Santa Monica, California: The RAND Corp., 1969.— 29 p.
51. Ford D. Shang inquiry as an alternative to Delphi: Some experimental findings // Technol. Forecasting and Social Change.— 1975.— 7, № 2.— P. 139—164.
52. Котляр С. Б. Модель коллективного принятия решения при локальном взаимодействии / VI Симпозиум по кибернетике, Тбилиси, 1—3 ноября, 1972.— Тбилиси: Ин-т кибернетики АН ГССР, 1972.— ч. 3.— С. 98—99.
53. Панкова Л. А., Шнейдерман М. В. Последовательная процедура экспертного опроса // Автомат. и телемех. 1975.— № 8.— С. 73—80.
54. Press S. J. Qualitative controlled feedback for forming group judgments and making decisions // J. American Statist. Assoc.— 1978.— 73, № 363.— P. 526—535.
55. Шнейдерман М. В. Экспертная процедура с индивидуальной обратной связью // II Всесоюзная конф. по статистическому и дискретному анализу нечисловой информации и экспертным оценкам.— М.: ВНИИ системных исследований.— 1984.— С. 394—395.
56. Шнейдерман М. В. Процедуры коллективного экспертного опроса и их экспериментальное исследование // Автомат. и телемех.— 1988.— № 5.— С. 3—16.
57. Шнейдерман М. В. Система многовариантной экспертизы для выбора параметров технологического процесса // Принятие решений и анализ экспертной информации.— М.: Науч. совет по комплексной проблеме «Кибернетика».— 1989.— С. 118—129.
58. Терехина А. Ю. Анализ данных методами многомерного шкалирования.— М.: Наука, 1986.— 168 с.
59. Терехина А. Ю. Представление структуры знаний методами многомерного шкалирования.— М.: ВНИИСИ, 1988.— 53 с.
60. Johnson S. C. Hierarchical clustering schemes // Psychometrika.— 1967.— 32.— P. 241—254.
61. Reitman J. S., Rueter H. H. Organization revealed by recall orders and confirmed by pauses // Cognitive Psychology.— 1980.— 12.— P. 554—581.
62. McKeithen K. B., Reitman J. S., Rueter H. H., Hirtle S. C. Knowledge organization and skill differences in computer programmers // Ibid.— 1981.— 13.— P. 307—325.
63. Kelley G. A. The psychology of personal constructs — N. Y.: Norton, 1955.— 311 p.
64. Boose J. H. A knowledge acquisition program for expert systems based on personal construct psychology // Int. J. Man-Mach. Stud.— 1985.— 23.— P. 495—525.
65. Boose J. H. Expertise transfer for expert system design.— N. Y.: Elsevier, 1986.
66. Shaw M. L., Gaines B. R. KITTEN. Knowledge initiation and transfer tools for experts and novices // Int. J. Man-Mach. Stud.— 1987.— 27.— P. 251—280.

моделей производственных систем и процессов, диалоговых процедур и средств машинной графики.

В третьем направлении представлены методы планирования, основанные на знаниях. Этот последний по времени возникновения, но, пожалуй, наиболее перспективный подход связан с созданием экспертных систем для календарного планирования. Здесь рассматриваются такие вопросы, как способы представления знаний для задач планирования, методы построения планов и расписаний с использованием знаний и процедур логического вывода, применение экспертных систем для календарного планирования.

1. ОБЩАЯ ФОРМУЛИРОВКА И КЛАССИФИКАЦИЯ ЗАДАЧ КАЛЕНДАРНОГО ПЛАНИРОВАНИЯ

В формулировке задач календарного планирования можно выделить две основные составляющие: задания* (широко используются также термины работы, заявки на обслуживание, виды деятельности, которые подлежат выполнению, и ресурсы, необходимые для организации выполнения заданий). Задания могут характеризоваться: а) директивными сроками начала и окончания выполнения; б) штрафными функциями, определяющими затраты, которые возникают при нарушении директивных сроков; в) отношением предшествования, которое для различных групп заданий определяет порядок их выполнения; г) различными этапами или фазами (стадиями) выполнения; д) длительностью выполнения в целом или отдельных фаз при различных уровнях обеспеченности ресурсами*; е) наличием полной или частичной упорядоченности различных фаз выполнения; ж) возможностью (или невозможностью) прерывания выполнения. Ресурсы по своему характеру могут быть разделены на следующие группы: а) единично-штучные ресурсы различных типов, в дальнейшем называемые процессорами, которые обязательно входят в набор ресурсов, обеспечивающих выполнение любого задания или его фаз, причем входят в этот набор в единственном числе; б) штучные ресурсы различных типов, которые входят в наборы обеспечивающих ресурсов определенными подмножествами**; в) потребляемые ресурсы различных типов, для каждого

* Вместо длительностей выполнения операций или их фаз могут задаваться функции интенсивности (скорости) выполнения. При этом должны задаваться объемы операций (фаз операций).

** Предполагается, что ресурсы двух первых групп являются непотребляемыми, т. е. могут использоваться по крайней мере в течение всего цикла выполнения заданий.

из которых заданы уровни предельного потребления в каждый момент времени и (или) за определенный период времени.

Задача календарного планирования — это по существу задача такого распределения во времени имеющихся ресурсов по исходным заданиям (или их фазам, если они выделены), при котором достигается экстремума некоторая целевая функция. Целевая функция, определяющая качество выработкиваемого календарного плана выполнения заданий, формализует, как правило, одну из следующих целевых установок:

1) быстродействие (минимум времени выполнения всех заданий);

2) минимум штрафных санкций (минимум суммы значений штрафных функций, определяемых отклонениями сроков выполнения заданий от директивных сроков);

3) минимум степени неравномерности использования потребляемых ресурсов;

4) минимальный простой процессоров и (или) некоторых других штучных ресурсов.

Приведенная формулировка задач календарного планирования весьма близка к формулировкам, приведенным в разделе I монографии [1] и в обзорах [2—14]. Известны различные классификации задач календарного планирования (см., в частности, [2, 4, 5]). Не вдаваясь в детали различных схем классификации, остановимся на основных ее признаках.

Первый признак: детерминированность — недетерминированность.

Второй признак: хорошая формализуемость — плохая формализуемость.

Третий признак: один процессор — много процессоров (во втором случае может использоваться дополнительный признак: однородные процессоры — неоднородные процессоры).

Четвертый признак: один критерий — много критериев (основные типы критериев были перечислены выше).

Дадим краткую характеристику первому и второму из выделенных признаков классификации. Третий и четвертый признаки вместе с другими традиционными признаками классификации задач календарного планирования детально рассмотрены в [2].

Детерминированными называются задачи календарного планирования, в которых к моменту начала разработки календарного плана имеется полная информация о всех заданиях, в том числе информация об их числе и директивных сроках возможного начала и обязательного завершения выполнения заданий* и полная информация о наличии и режимах использования ресурсов. В недетерминированных задачах кален-

* Заметим, что в ряде случаев потребность в выполнении некоторых заданий может возникать периодически (см., например, [6]), но в детерминированных задачах число периодов должно быть априори известно.

дарного планирования исходное множество заданий априори не определено и по мере реализации частично составленного календарного плана множества заданий может пополняться в реальном времени и новые задания должны включаться в формируемое расписание. Пополнение множества исходных заданий осуществляется случайным образом в соответствии с известным или гипотетическим распределением, или в условиях отсутствия каких-либо вероятностных закономерностей (иногда говорят, о неопределенности невероятностного характера)*. Недетерминированность заданий может быть также связана с наличием вероятности достижения не полного результата при выполнении задания (полный или частичный брак в результате случайных внешних воздействий и др.). Наконец, элемент недетерминированности в задаче календарного планирования может вносить ненадежность ресурса, который может становиться в случайные моменты времени недоступным для использования.

Уровень формализуемости задач календарного планирования в основном связан: а) с возможностью более или менее точного задания зависимости времени, или интенсивности, выполнения задания от выделенных для этого наборов ресурсов; б) с наличием более или менее четкой целевой установки и с возможностью ее формализации; в) со степенью размытости задаваемых директивных сроков и предельных уровней используемых потребляемых ресурсов; г) с наличием неопределенности в сопоставлении различных целевых установок (критериев), фигурирующих в задаче. Если при формальном описании задачи календарного планирования перечисленным пунктам удается дать однозначное толкование, адекватное исходной практической ситуации, то рассматриваемую задачу можно считать хорошо формализуемой. При невозможности получения такого однозначного описания по любым из перечисленных пунктов соответствующую задачу следует считать плохо формализуемой.

Математические модели, описывающие задачи календарного планирования, весьма разнообразны и носят дискретный характер, поскольку связаны с распределением во времени частично дискретных ресурсов по неделимым заданиям или их фазам. Среди разделов прикладной математики, «обслуживающих» календарное планирование, следует, в первую очередь, назвать теорию расписаний [1, 7, 8], а также комбинаторное программирование [9, 10, 11] и теорию графов [12, 13, 14].

Задачи календарного планирования возникают в целом ряде прикладных областей. Среди основных следует назвать область

* Иногда задачи, в которых формирование календарного плана проводится в реальном времени, при вероятностном или произвольном характере пополнения множества заданий во времени, называют динамическими задачами календарного планирования в отличие от детерминированных задач, называемых в этом случае статическими [5].

организации производственных и вычислительных процессов [1, 4, 5, 7, 8], область организации работы транспорта [3, 15], область разработки расписаний учебных занятий (см., например, [16, 17]). При этом в области организации производственных процессов можно выделить весьма специфическую подобласть — управление запасами [18]. Кроме того, весьма близки к задачам календарного планирования, возникающим при организации производственных (в условиях вероятностной неопределенности) и вычислительных процессов задачи массового обслуживания [19, 20].

В силу большого разнообразия задач календарного планирования весьма разнообразны и методы их решения. Существующие методы можно разбить на две основные группы: методы, ориентированные на решение хорошо формализуемых детерминированных задач, и методы, ориентированные на решение недетерминированных и (или) плохо формализуемых задач. Что касается методов первой группы, то поскольку в большинстве случаев речь здесь идет о решении *NP*-полных задач, то основную роль в этой группе играют приближенные эвристические методы. Такие методы описаны в большом количестве работ (систематическое описание части таких методов проводится в [21]). В следующем разделе в рамках основных подходов к разработке приближенных эвристических методов календарного планирования, рассмотрены новейшие результаты в данной области.

Что касается методов второй группы, то они основаны на использовании имитационного моделирования и идеологии экспертных систем. Новейшие результаты в этой области рассмотрены в разделах 3 и 4 настоящего обзора.

2. ПРИБЛИЖЕННЫЕ МЕТОДЫ РЕШЕНИЯ ДЕТЕРМИНИРОВАННЫХ И ХОРОШО ФОРМАЛИЗОВАННЫХ ЗАДАЧ КАЛЕНДАРНОГО ПЛАНИРОВАНИЯ

Методы, использующие разрешимые частные случаи. Уточним, что разрешимым частным случаем некоторой задачи комбинаторного программирования принято называть такой ее частный вариант, который выделяется эффективно проверяемыми условиями, при этом существуют эффективные точные алгоритмы решения. Под эффективными подразумеваются алгоритмы с полиномиальной относительно размеров задачи оценкой трудоемкости.

Допустимы три основные схемы, использующие разрешимые частные случаи. Первая схема: разбиение задачи на фрагменты, представляющие собой некоторые разрешимые частные случаи; нахождение с помощью эффективных алгоритмов оптимальных решений задач — фрагментов; построение из получен-

ных решений приближенного (реже точного) решения исходной задач. Вторая схема: минимальная корректировка набора данных исходной задачи, приводящая ее к некоторому разрешимому частному случаю; построение с помощью эффективного алгоритма оптимального решения; принятие этого решения в качестве приближенного решения исходной задачи. Третья схема: выбор одного или нескольких разрешимых частных случаев; генерация K наилучших решений в каждом из разрешимых частных случаев, где K — некоторый целочисленный параметр; выбор наилучшего из всего множества сгенерированных решений.

Рассмотрим примеры реализации каждой из трех названных схем. В качестве первого примера рассмотрим одну из классических задач календарного планирования. Имеется N заданий, готовых к выполнению в некоторый фиксированный начальный момент времени. Для заданий определены директивные сроки D_j желательного завершения их выполнения ($j = \overline{1, N}$). Имеется M идентичных процессоров, на которых время выполнения j -го задания определяется величиной t_j . Будучи начато, выполнение заданий не прерывается. Необходимо так распределить задания по процессорам и выбрать такой порядок их выполнения на каждом из процессоров, чтобы в полученном расписании (календарном плане) число работ, не законченных к соответствующим срокам D_j , было минимально. Для одного процессора эта задача может быть эффективно решена [22]. Чтобы использовать этот разрешимый частный случай в рамках первой схемы, упорядочим задания в соответствии с величинами D_j или t_j так, как это показано на рис. 1а или на рис. 1б. Далее множество всех заданий разобьем на M групп. Если упорядоченную таким образом последовательность обозначить через (j_1, j_2, \dots, j_N) , то в первом случае разбиение получается путем разрезания этой последовательности на M «кусков» по $m = ([N/M] + 1)$ элементов в каждом (кроме, быть может, последнего, который содержит $(N - m(M - 1))$ элементов). Во втором случае формирование M групп осуществляется следующим образом: 1-я группа — $(j_1, j_{M+1}, j_{2M+1}, \dots)$; 2-я группа — $(j_2, j_{M+2}, j_{2M+2}, \dots)$ и т. д. Отметим, что поскольку можно упорядочивать как величины D_j , так и величины t_j , причем упорядочивать двумя различными способами, можно получить (упорядочивая оптимальным образом задания каждой группы) четыре решения, лучшее из которых и следует принять за приближенное решение рассматриваемой задачи.

В качестве второго примера рассмотрим задачу обработки N заданий на M процессорах при следующих специфических условиях: все процессоры идентичны; отсутствуют директивные сроки завершения заданий; все задания в начальный момент предъявлены к выполнению; заданы ограничения предшествования для заданий, которые могут быть представлены в виде

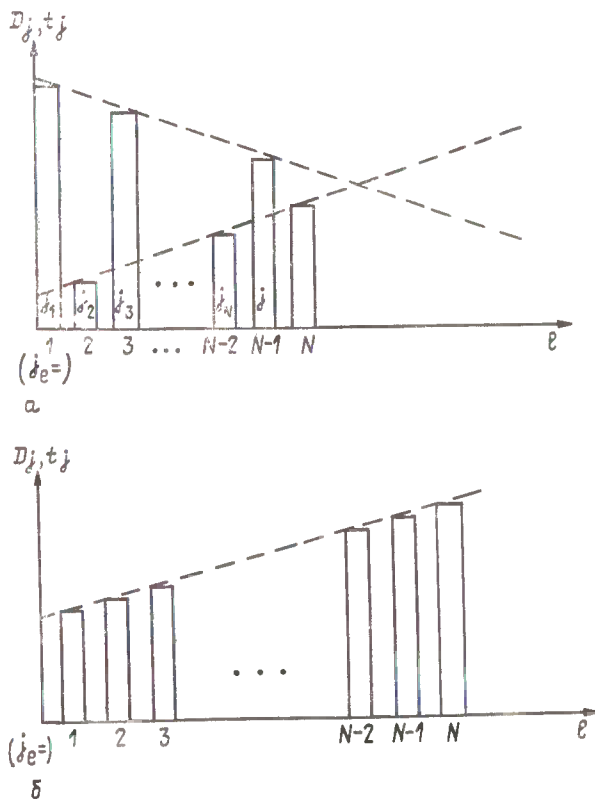


Рис. 1. Варианты упорядочения заданий

ориентированного дерева или леса; качество расписания оценивается либо максимальным временем выполнения всей программы, либо взвешенной суммой времен выполнения отдельных заданий. Данная задача может быть эффективно решена в случае, если времена выполнения заданий на процессорах одинаковы (см., например, раздел 1.4 в [1], а также [23]). С учетом этого, в рамках второй схемы следовало бы от исходной задачи перейти к задаче с одинаковым временем t_0 выполнения заданий на процессорах, где t_0 можно задать одним из следующих двух способов:

$$t_0 = 1/N \sum_{j=1}^N t_j, \quad t_0 = t_{\min} + \frac{1}{2}(t_{\max} - t_{\min}), \quad (1)$$

где $t_{\min} = \min_j \{t_j\}$, $t_{\max} = \max_j \{t_j\}$.

Решение задачи с одинаковыми временами выполнения заданий, равными t_0 , дает тем лучшее приближенное решение исходной задачи, чем меньше величина

$$\Delta t = \max\{t_{\max} - t_0, t_0 - t_{\min}\}.$$

Наконец, третья схема использования разрешимых частных случаев может быть продемонстрирована на том же втором примере. В рамках этой схемы следует, выбрав с помощью (1) значение t_0 , рассмотреть частный случай задачи с одинаковыми временами выполнения заданий на процессорах. В данном случае следует искать не единственное оптимальное решение этого частного случая, а некоторое подмножество наилучших решений. Можно это проделать для различных значений t_0 . Далее из полученных решений следует выбрать наилучшее приближенное решение исходной задачи. Центральным моментом при реализации третьей схемы является построение подмножества наилучших решений. Такое построение может оказаться достаточно сложным даже для эффективно решаемых задач. Некоторые общие подходы к построению подмножеств наилучших решений для эффективно решаемых задач календарного планирования рассмотрены в [24].

Методы, основанные на схемах динамического программирования. Идея динамического программирования состоит в погружении исходной задачи в параметрически задаваемое семейство задач сходной структуры и организации на этом семействе последовательного решения задач. Метод эффективен, когда удастся выбрать семейство задач и задать последовательность их решения таким образом, что результаты любого промежуточного этапа в существенной мере используют результаты предыдущего этапа.

Как правило, при построении семейств задач используют один или несколько характерных параметров задачи, значения которых варьируются в пределах, обеспечивающих, с одной стороны, достаточно эффективное решение задачи для начального значения параметра n , с другой стороны, вхождение в рассматриваемое семейство исходной задачи. При решении методом динамического программирования различных подклассов задач календарного планирования в качестве характеристик параметров могут выступать число заданий N , число процессоров M и число порций используемого ресурса. В зависимости от того, какой из этих параметров используется при формировании семейства вспомогательных задач, различают три схемы динамического программирования. В первой схеме рассматривается семейство задач, охватывающих всевозможные подмножества $I_n = \{j_1, j_2, \dots, j_n\}$ заданий. Отдельная задача представляет собой фрагмент исходной задачи с соответствующим подмножеством заданий I_n . Решаются задачи семейства в порядке монотонного возрастания целочисленного параметра n (определяющего число работ), изменяющегося в интервале $[1, M]$, причем, увеличение n происходит только после решения задач со всевозможными подмножествами заданий I_n . Заметим, что в некоторых задачах календарного планирования удастся при использовании данной схемы динамического программирования для каж-

дого фиксированного n решать весьма незначительную часть задач семейства. В частности, это удастся делать при решении задач, обобщающих классическую задачу Джонсона [25]. Заметим, что первая схема применима к решению однопроцессорных задач календарного планирования или многопроцессорных задач, в которых маршруты прохождения заданий по процессорам одинаковы.

В отличие от первой вторая схема динамического программирования применяется для решения многопроцессорных задач календарного планирования. В этой схеме в отдельных задачах семейства проводится оптимальное разбиение подмножеств I_n на m групп, где $m \leq n$ и отдельные группы заданий «привязаны» к определенным процессорам. Решение задач проводится в порядке монотонного изменения параметров m и n в интервалах соответственно $[1, \min\{n, M\}]$ и $[1, M]$.

Наконец, третья схема, может быть продемонстрирована на следующем примере. Пусть имеется N заданий, выполняемых на одном процессоре. Время выполнения заданий τ может меняться в зависимости от представляемого дополнительного ресурса r . Соответствующие зависимости $\tau_j(r_j)$ являются выпуклыми и равномерно дискретными по времени. Максимальный суммарный объем потребления ресурса r фиксирован. Требуется так распределить ограниченный ресурс по заданиям, чтобы общее время выполнения программы было минимальным. Решение данной задачи по третьей схеме осуществляется в итерационной процедуре последовательного добавления в систему определенных порций ресурса, увеличивающих общее время на один интервал дискретности зависимостей $\tau(r)$. При этом однозначно определяются величина добавляемой порции ресурса и задание, в «поддержку» которого выделяется эта порция. Детально намеченная процедура динамического программирования описана в разделе 3 монографии [26], где она применяется и для решения более общих задач.

Гриды-методы. Гриды-методы, как правило, реализуются в виде итерационных процедур, в ходе которых последовательно фиксируются определенные элементы решения. При такой фиксации придерживаются недальновидной стратегии получения максимального приращения функционала на каждом шаге. Поэтому соответствующие алгоритмы и названы гриды-алгоритмами (greedy), что в переводе с английского означает «пожирающие», или «жадные» алгоритмы.

В решении задач календарного планирования применяются две основные гриды-схемы. В рамках первой схемы априори для всех заданий или для некоторого подмножества заданий, составляющих текущий фронт, определяются приоритеты очередности выполнения. Приоритеты определяются использованием некоторых функций, аргументами которых могут быть параметры, оценивающие и (или) прогнозирующие качество

формируемого расписания при постановке в определенное место этого расписания соответствующего задания. Предложены самые разнообразные приоритетные функции. Наиболее типичные из них рассмотрены в [1, 20, 27].

Вторая схема построения гриди-алгоритмов может быть продемонстрирована на задаче составления расписания N независимых заданий на M процессорах. На каждой итерации соответствующей процедуры из числа не включенных в расписание заданий и из числа процессоров с недоиспользованным ресурсом (например, ресурсом времени) выбирается наилучшая пара «задание — процессор». Причем качество указанной пары оценивается приращением функционала при включении выполнения соответствующего задания в расписание работы соответствующего ему процессора. В соответствии со сделанным выбором пары «задание — процессор» осуществляется корректировка текущего расписания. Существует менее «жадная» модификация описанной схемы, в которой очередная выбранная пара «задание — процессор» не корректирует сразу текущее расписание, а лишь включается в множество пар-кандидатов, определяющих в последующем соответствующую корректировку. Причем некоторая пара-кандидат становится корректирующей после появления во множестве кандидатов альтернативной ей пары. Более детально модифицированная гриди-схема описана в применении к некоторым подклассам задач группировки в [26, разделы 5, 6] и в [28].

Методы локальной оптимизации. Рассматриваемые в данном разделе методы, как правило, реализуются путем задания некоторого множества допустимых локальных изменений исходного решения и выбора в этом множестве такого изменения, которое приводит к наилучшему решению (иногда к любому решению, лучшему исходного). Алгоритмы локальной оптимизации являются итерационными. При этом решение, полученное на множестве допустимых локальных изменений в текущей итерации, становится начальным в следующей итерации. Признаком окончания работы алгоритмов локальной оптимизации является невозможность получения на какой-либо итерации (эта итерация становится последней) решения, лучшего начального для этой итерации решения. По такой классической схеме могут быть построены и строятся алгоритмы локальной оптимизации календарных планов. В качестве операторов локальной оптимизации, задающих множество локальных изменений некоторого фиксированного расписания в такой классической схеме, может использоваться оператор $P_1(k, l)$, заданный на множестве Π_N всех перестановок $\pi = (\pi_1, \pi_2, \dots, \pi_N)$ из элементов множества $I_N = \{1, 2, \dots, N\}$ (представляющего множество связей), и оператор $P_2(I_k, I_l, n_k, n_l)$, заданный на множестве разбиений $I_{N, M} = \{I_1, I_2, \dots, I_M\}$ множества I_N на M непересекающихся подмножеств I_i ($i = \overline{1, M}$, где M — число процессоров). Опера-

тор $P_1(k, l)$ осуществляет замену местами в перестановке π элементов, занимающих позиции k и l ($k, l = \overline{1, N}$). При этом перестановка π задает очередность выполнения заданий в начальном расписании. Оператор $P_2(k, l, I_k, I_l)$ осуществляет замену местами каких-либо n_k элементов из подмножества I_k на какие-либо n_l элементов из подмножества I_l . При этом разбиение $I_{N, M}$ задает распределение N заданий по M процессорам в начальном расписании. Следует отметить, что при использовании оператора P_2 оценка приращения функционала может учитывать изменение очередности выполнения заданий, входящих в скорректированные подмножества I_k и I_l .

Как показывает опыт практического применения алгоритмов локальной оптимизации в решении задач календарного планирования наиболее эффективны (по соотношению между получаемым приращением в «качестве» решения и трудоемкостью) алгоритмы, в которых множество локальных изменений имеет сравнительно небольшую мощность, оцениваемую полиномом от характерных параметров задачи (в нашем случае от N и M). Операторы P_1 и P_2 при n_k и n_l , не превышающих нескольких единиц этому условию вполне удовлетворяют.

В последнее время стали использоваться алгоритмы локальной оптимизации, в которых, в отличие от классических алгоритмов, игнорируется требование монотонности. В первой немонотонной схеме локальной оптимизации требование получения на каждой итерации решения, лучшего начального, заменяется требованием получения просто лучшего решения, которое может быть и хуже начального. Реализации такого рода схемы для некоторых задач календарного планирования приведены в [29], где указаны также некоторые приемы ликвидации заикливания, которое может иметь место при использовании данной схемы. Во второй немонотонной схеме локальной оптимизации соответствующая процедура переходов от одних решений к другим строится на аналогии с процессом отжига металла (simulated annealing). Одна из реализаций такой схемы применительно к решению задач календарного планирования, сводящихся к задаче упаковки, приведена в [30]. Другие реализации относятся к решению задачи коммивояжера, к которой, как показано в [30], можно «приближенно свести» некоторые задачи календарного планирования.

Методы декомпозиции. Рассматриваемый метод применяется к многопроцессорным задачам календарного планирования. Метод строится по схеме, включающей в себя три этапа.

Этап 1. Формируется агрегированная целевая функция исходной задачи. Агрегирование проводится таким образом, чтобы исключить явную зависимость целевой функции от очередности выполнения заданий на отдельных процессорах.

Этап 2. Решается задача оптимального разбиения множества всех заданий на подмножества, привязанные к отдельным

процессорам. При этом в качестве функционала используется агрегированная целевая функция, полученная на предыдущем этапе, а число подмножеств разбиения совпадает с числом процессоров M .

Этап 3. Решаются M (по числу сформированных на этапе 2 подмножеств I_i) задач упорядочения заданий на отдельных процессорах. Функционалы отдельных задач определяются как соответствующие «проекции» функционала исходной задачи.

Возможные реализации описанной схемы декомпозиции приведены в [26, раздел 6].

Методы векторной оптимизации. Известны четыре основные схемы, по которым строятся методы векторной оптимизации.

Первая схема реализует так называемый метод идеального решения (в литературе принято называть «методом идеальной точки»). Метод состоит в многократном независимом решении исходной задачи при учете каждый раз только одного из заданных критериев. Таким образом, если число критериев в задаче равно \bar{K} , то в рамках рассматриваемого метода должны быть получены K решений R_k ($k=1, \bar{K}$), наилучших по соответствующим критериям. Если R_k ($k=1, \bar{K}$) совпадают, то, естественно, решение исходной задачи найдено. В противном случае в пространстве решений вводится метрика. С использованием этой метрики решается задача построения такого решения R_0 , для которого минимально суммарное отклонение от решений R_k ($k=1, \bar{K}$). Говоря о суммарном отклонении, мы имеем в виду фактическую сумму отклонений или максимальное из отклонений решения R_0 от решений R_k ($k=1, \bar{K}$).

Вторая схема реализует метод свертки критериев. Метод состоит в построении по K заданным критериям одного критерия, в той или иной мере учитывающего исходные критерии. Такое преобразование нескольких критериев в один обычно называют сверткой критериев. Во многих случаях приемлемые результаты могут быть получены для достаточно простых приемов свертки (в частности, при «адаптивной» свертке критериев).

Третья схема реализует метод замещения критериев дополнительными ограничениями. Метод состоит в задании для отдельных критериев уровней и в замене критериев условиями, выполнение которых гарантирует достижение этих уровней. Применение данного метода позволяет уменьшить число критериев и снизить тем самым сложность решения исходной задачи.

Четвертая схема предполагает выделение подмножества решений, которые в некотором смысле доминируют над остальными решениями. Одним из наиболее распространенных типов множеств доминирующих решений является так называемое множество Парето. Как правило, множество Парето включает относительно большое число возможных решений и тре-

буется и требуется использовать специальные приемы, чтобы сократить его до практически приемлемых размеров.

Методы, реализующие первые три описанные схемы, очевидным образом сводятся к методам, рассмотренным выше, при условии, что рассматриваемая задача является «многокритериальным» обобщением задач, упомянутых в соответствующих разделах. Что касается четвертой схемы, то она, по-видимому, может быть реализована лишь в рамках переборных процедур, построенных, например, по схеме ветвей и границ. Методы ветвей и границ в данном обзоре не рассматриваются. Тем не менее, можно указать пример реализации схемы доминирования в рамках процедуры ветвей и границ, описанный в [26, раздел 6].

Четыре описанные схемы векторной оптимизации имеют достаточно длительный и широкий опыт практического применения. Сравнительно недавно эти классические схемы были дополнены новой, пятой схемой, в рамках которой, в частности, возможна генерация множества доминирующих решений (см. описание схемы 4), и вообще, реализация всех классических схем [24].

Пятая схема предполагает, что имеется возможность генерации по каждому критерию подмножеств наилучших решений. Генерация наилучших решений проводится либо до получения заданного числа (по каждому критерию) наилучших решений, либо до получения решения, на котором происходит «скачкообразное изменение» качества решений. После генерации полученные подмножества решений объединяются, после чего объединенное множество решений анализируется с использованием одной из классических схем.

Заметим, что применение последней схемы возможно, как правило, для достаточно узких подклассов задач календарного планирования. Некоторые из этих подклассов указаны в [24].

3. КАЛЕНДАРНОЕ ПЛАНИРОВАНИЕ НА БАЗЕ ИМИТАЦИОННЫХ МОДЕЛЕЙ

Ограничения, присущие рассмотренным в разделе 2 формальным моделям и методам календарного планирования, и неудачи, часто сопутствующие попыткам их применения в реальных производственных условиях, породили стремление использовать при решении практических задач другой, более прагматический подход. Он состоит в организации процедур календарного планирования на базе имитационного моделирования (ИМ). Построение расписания здесь осуществляется в ходе интерактивного взаимодействия пользователя — лица, принимающего решения (ЛПР) с программной имитационной моделью производственной системы, которая должна быть построена (или настроена) предварительно. Об оптимальности получаемых расписаний говорить здесь, как правило, не прихо-

дится. В общем виде процедура построения расписания состоит из следующих шагов:

- 1) производится загрузка исходных данных,
- 2) ЛПР задает способ (один или несколько) построения расписания и желаемые показатели его качества,
- 3) в соответствии с заданием система с помощью модели генерирует один или несколько вариантов расписания и вычисляет показатели их качества,
- 4) ЛПР сравнивает эти варианты и либо выбирает один из них в качестве окончательного, после чего выполняется п. 5, либо изменяет исходное задание, корректируя входные данные, выбирая другой способ построения расписания или другие показатели качества, после чего вновь выполняется п. 3.
- 5) система выдает готовое расписание в удобной для ЛПР форме.

Варианты расписания на шагах 2 и 3 генерируются путем просчета имитационной модели с использованием приоритетных правил или эвристических алгоритмов, известных в теории расписаний. Поскольку последние уже были рассмотрены в разделе 2, здесь мы коснемся средств ИМ, являющегося основным инструментом в рамках данного подхода. К настоящему времени методология ИМ, в т. ч. ИМ производственных систем, сформировалась уже достаточно четко [31—33]. Модель производственной среды должна с достаточной степенью детализации отражать ее актуальное состояние, допускать удобную модификацию при изменении отдельных ее элементов и обеспечивать удобное взаимодействие с ЛПР в процессе планирования. С целью облегчения и ускорения построения имитационных моделей разработано большое число специальных программных систем, называемых обычно «языками имитационного моделирования». В рамках своей основной цели — облегчить формализованное описание моделируемой системы языка ИМ обеспечивают обычно выполнение следующих функций [31]: структурированное описание входных данных, отсчет системного времени, стандартный вывод результатов моделирования, диагностику ошибок, генерирование случайных величин и др. Языки ИМ делятся на три группы: дискретного, непрерывного и комбинированного моделирования, причем эти термины связаны, в первую очередь, с характером получаемых моделей, а не моделируемых систем. Для задач календарного планирования строятся, как правило, дискретные модели и используются соответственно языки дискретного или комбинированного ИМ. Наиболее известными языками дискретного ИМ являются GPSS, SIMSCRIPT, GASP II; комбинированного — SIMAN, SLAM II. Обзор описательных средств, используемых в языках ИМ, можно найти в [31]. Наряду с языками ИМ общего назначения разрабатываются и специальные средства ИМ производственных систем [35].

В настоящее время большое число систем ИМ, в т. ч. и все названные языки ИМ, реализовано на персональных ЭВМ; обзор этих программных средств приведен в [34].

По мере распространения интерактивного режима общения с ЭВМ быстро развивались средства, которые обеспечивают удобное взаимодействие пользователя с моделью. Постепенно они выделялись в самостоятельные компоненты. К ним относятся, во-первых, диалоговые средства управления моделью (диалоговые мониторы), поддерживающие функции задания вариантов, взаимодействия с терминалом, вывода результатов. Во-вторых, это приобретающие все более важное значение средства графического отображения информации, которые либо входят в состав диалоговых мониторов, либо оформляются в виде самостоятельных пакетов машинной графики.

Развитие традиционных программных средств ИМ идет в двух основных направлениях [36]. Первое связано с дальнейшим совершенствованием языков и систем ИМ. Но при всех достижениях в этом направлении функция создания модели остается все же за пользователем. Второе направление заключается в создании генераторов моделирующих программ, которые переводят описанную пользователем в достаточно общем символьном виде логику модели в коды языка ИМ. Такие генераторы позволяют практически снять с пользователя трудности по созданию модели. Примерами традиционного подхода к применению ИМ могут служить система AMPS, а также коммерческие системы SCHED/SIM и SCHEDULEX.

Система AMPS [97] представляет собой интеллектуальную систему поддержки моделирования дискретных производственных систем, которая помогает пользователю в разработке модели исследуемой производственной системы и автоматизирует составление программы моделирования на языке GPSS/PC (версия языка GPSS для персональных ЭВМ класса IBM PC XT/AT). Однако собственно построения календарных планов система AMPS не производит.

Система SCHED/SIM [38], будучи достаточно простой, для генерации расписаний использует только эвристические правила приоритета. В системе SCHEDULEX [37] расписания строятся с помощью набора оптимизационных программ. Для создания модели производственной среды не требуется кодирования и программирования — пользователь строит модель, пользуясь выводимыми на экран образцами таблиц. Предусмотрена возможность создания и использования иерархических моделей: сначала строится расписание на верхнем уровне, которое затем детализируется на нижних уровнях. Система SCHEDULEX реализована на персональных ЭВМ класса IBM PC.

В [39] описывается использование детерминированной событийной модели для составления графика функционирования установки по переработке партий плазмы человеческой крови.

Процесс обработки состоит из четырех основных фаз, одновременно может обрабатываться до 10 партий плазмы с различными технологическими требованиями. Модель реализована на языке SLAM II на универсальной ЭВМ семейства IBM. Составление допустимого графика на 6 недель занимает 10—15 с. Процедура построения графика состоит в имитации применения диаграммы Ганта.

Одной из разновидностей рассматриваемого подхода ряд авторов считает т. н. визуальное ИМ [40], возникшее в рамках исследования операций и использующее развитые средства машинной графики для отображения моделей и общения ЛПР с ними. Система визуального ИМ содержит три основных компонента: 1) визуальная модель задачи планирования (или анализируемой системы), 2) процедуры, дающие сравнительно приемлемые варианты расписания, отображаемые затем с помощью визуальной модели, 3) «дружественная» интерактивная среда моделирования. Системы визуального ИМ особенно полезны в тех случаях, когда ЛПР имеет четкий зрительный образ расписания или объекта планирования (например, в виде диаграммы Ганта, графика работы персонала, схемы дорожной сети). Обзор работ по визуальному ИМ приводится в [40].

Новым направлением в ИМ является т. н. интеллектуальное моделирование [36, 91, 92] — гибридный подход, основанный на взаимодействии средств ИМ и ЭС. Классификация возможных способов построения интеллектуальных систем ИМ приводится в [91]. Рассмотрение этого вопроса выходит за рамки настоящего обзора, укажем только на самый простой способ «интеллектуализации» ИМ — реализацию модели производственной системы на языке логического программирования ПРОЛОГ или ПРОЛОГ-подобном [93]. Такой подход применительно к составлению расписаний в гибкой производственной системе рассматривается в [94]. Основное достигаемое при этом преимущество состоит в отделении описания логики функционирования системы от описания способа управления ею, что облегчает использование модели при различных стратегиях поиска расписания по сравнению с традиционными процедурно-ориентированными языками ИМ (GPSS, SLAM II, SIMSCRIPT и др.), которые требуют детального описания функционирования системы, включая и точную процедуру планирования.

4. КАЛЕНДАРНОЕ ПЛАНИРОВАНИЕ НА БАЗЕ МЕТОДОВ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА

4.1. Проблема планирования с позиций искусственного интеллекта

Проблема планирования последовательности действий зародилась в области искусственного интеллекта (ИИ) в 60-е годы в рамках общей проблемы автоматического решения задач и

восходит к известному «универсальному решателю задачи» [41, 42]. Ряд специалистов по ИИ до сих пор трактуют эти понятия как синонимы. Традиционно предметом исследования здесь являлись задачи планирования действий в широком смысле (иногда называемые «классическими» [42, 43]), которые состоят в следующем: полагая заданными 1) совокупность целей (множественность целей характерна для этих задач), 2) множество допустимых действий, 3) описание начальной ситуации, найти некоторую последовательность действий (план), приводящую систему в такую конечную ситуацию, при которой достигаются все поставленные цели. В такой постановке задача планирования имеет очень широкие приложения. Планы как некоторая базовая структура данных могут быть использованы в таких областях, как планирование действий роботов, автоматическое программирование ЭВМ, синтез естественного языка и др. Очевидно, что и задачи календарного планирования вполне укладываются в такую постановку.

Следует отметить существенное различие подходов к проблеме планирования с позиций ИИ, с одной стороны, и с позиций исследования операций и теории принятия решений, с другой. В первом случае это построение некоторой части пространства планов, с тем чтобы найти планы, удовлетворяющие желаемым критериям. Во втором случае это поиск оптимального по некоторому критерию плана или выбор наилучшего плана среди небольшого числа полностью определенных альтернатив.

Проблема планирования в широком смысле по сравнению с другими традиционными направлениями исследований в ИИ, такими, как эвристический поиск, методы логического вывода, машинное обучение и др., имеет ряд специфических особенностей, к числу которых можно отнести [44]: 1) представление знаний и рассуждения а) о времени, б) о причинности, в) о намерениях; 2) наличие неопределенности, связанной с исполнением плана в «реальном мире»; 3) восприятие «реального мира»; 4) физические и другие ограничения на допустимые решения; 5) возможное наличие нескольких агентов, которые могут сотрудничать или противодействовать друг другу. Видно, что некоторые из этих особенностей, в т. ч. указанные в п. 1а, 1б, 2, 4, свойственны и задачам календарного планирования.

Исторически исследования по автоматическому планированию в рамках ИИ развивались в двух направлениях: 1) создание универсальных, основанных на знаниях систем, генерирующих планы действий автоматически и не привязанных к области применения, 2) разработка специализированных ЭС для конкретных прикладных областей.

4.2. Автоматические системы построения планов

Автоматические планировщики разрабатывались обычно с целью решения классических задач планирования. Традицион-

ная формальная схема таких задач предполагает, что окружающий мир может находиться в одном из потенциально бесконечного числа состояний. Под влиянием любого действия мир переходит из одного состояния в другое, поэтому модель действия представляет собой совокупность пар «текущее состояние — следующее состояние», описывающих, как данное действие изменяет текущее состояние окружающего мира.

Большинство планировщиков, не привязанных к области применения и разработанных преимущественно до середины 70-х годов, построено по принципу планирования в пространстве состояний [45] или реализует незначительные расширения этого подхода. Данный подход возник на базе ситуационного исчисления и состоит в следующем. Пусть заданы множество условий, описывающих начальное состояние, и множество условий, заданных в качестве цели. Нужно найти такую последовательность действий (операторов), которая, будучи применена к любой ситуации, удовлетворяющей начальным условиям, породит ситуацию, удовлетворяющую целевым условиям. При этом предполагается, что будущее однозначно определяется начальной ситуацией и последовательностью операторов, выполняемых, начиная с этой ситуации. Для описания факторов, касающихся состояний мира, целей плана, выполняемых действий, используется обычно исчисление предикатов первого порядка.

Многие результаты в рамках этого подхода получены при разработке автоматических планировщиков действий роботов. Планирование действий роботов — наглядный пример классической задачи планирования. В типичной задаче такого рода рассматривается робот, способный выполнять определенный набор операций в некотором легком для понимания мире, например перемещать кубики в «мире кубиков» или передвигаться и перемещать некоторые предметы в помещении, содержащем другие предметы. Эти традиционные задачи часто используют для лучшего понимания и сравнения возможностей различных методов планирования, хотя более современные системы планирования вряд ли демонстрируют свои подлинные возможности на таких простых задачах.

Родоначальником подобных систем был планировщик STRIPS [46], создававшийся под влиянием «универсального решателя задач» [41]. Основной вклад планировщика STRIPS — это т. н. STRIPS-допущение при решении проблемы рамок и формальный язык для описания действий (проблема рамок заключается в том, как, описывая последствия отдельных действий, выразить, что каждое действие преобразует, как правило, лишь малую часть окружающего мира). В системе STRIPS действия робота моделируются посредством правил (операторов), изменяющих одно состояние мира на другое, а последствия каждого действия указываются в явном виде списками вычеркиваний и добавлений. Принятая форма операторов содержит три

компоненты. Первой является формула предварительного условия (предусловия), которая определяет применимость данного правила к определенному состоянию. Вторая компонента — это список вычеркиваний, содержащий те элементы, которые удаляются из описания прежнего состояния как первый шаг в конструировании нового состояния.

Третьей компонентой является список добавлений, содержащий те элементы, которые добавляются в описание предыдущего состояния (после выполнения вычеркиваний), чем и заканчивается конструирование нового состояния.

Вслед за системой STRIPS в 70-е годы был разработан ряд сходных планировщиков, в которых также использовался с теми или иными модификациями принцип планирования в пространстве состояний (ABSTRIPS, NOAH, NONLIN и др.). Среда планирования в STRIPS и STRIPS-подобных системах накладывает ряд существенных ограничений на спектр решаемых задач и качество планов. Это прежде всего обязательная «смежность» причины и следствия (последствия действий проявляются в следующем же состоянии); невозможность построения планов, содержащих совмещенные по времени действия, а также планов, исполнение которых начинается после того момента, для которого задана начальная ситуация (такие планы здесь не имеют смысла); невозможность решения задач, в которых мир меняется не только под воздействием исполнителя плана, но и в результате внешних воздействий; дискретность времени, не позволяющая рассматривать непрерывные процессы.

Более современная трактовка процесса поиска плана состоит в том, что точками в пространстве поиска являются не состояния мира, а некоторые частичные планы. Переход из одной точки в другую осуществляется путем выполнения любой допустимой операции планирования, например более детальное описание некоторого элемента плана, учет дополнительного ограничения и др.

Основная часть исследований по созданию автоматических планировщиков в рамках ИИ носила преимущественно экспериментальный характер и ставила целью продемонстрировать предлагаемые идеи в виде программ на ЭВМ. Задача построения плана трактовалась прежде всего как процедура эвристического поиска, при этом доминировали «слабые», т. е. бедные знаниями методы, когда пространство, в котором ведется поиск плана, строится при помощи сравнительно небольшого числа операторов. Цель создания каких-либо математических методов синтеза планов не ставилась (первые работы такого рода появились лишь в самые последние годы, например [43]), а усилия концентрировались на таких вопросах, как совершенствование стратегий управлений поиском*, использование доступной пла-

* Заметим, что некоторые методы поиска в пространстве состояний идентичны методам, которые в области исследования операций принято называть «методами ветвей и границ».

нировщику информации (знаний) о предметной области для ограничения пространства поиска, разработка средств для создания более реалистичных моделей прикладных областей, обеспечение более активного взаимодействия с пользователем. Примерно к середине 80-х годов появляется термин «планирование, основанное на знаниях» (knowledge based planning) [44], отражающий всевозрастающую роль знаний о прикладной области при построении планов автоматическими планировщиками.

Рассмотрим кратко лишь некоторые основные идеи и методы.

Иерархическое планирование. При этом подходе общая задача планирования решается на нескольких уровнях абстракции (в нескольких абстрактных пространствах), где используются агрегированные данные. В процессе перехода на более низкие уровни происходит постепенная детализация плана. При этом на каждом уровне может применяться любой из приемлемых одноуровневых алгоритмов поиска. Иерархическое планирование представляет собой по существу процедуру нисходящего проектирования. Планы, получаемые в процессе иерархического планирования, редко бывают оптимальными, однако решать сложные задачи, не прибегая к иерархическому подходу, практически невозможно. Показано [47], что если размерность исходного пространства задачи равна n , то оптимальная абстрактная иерархия для нее содержит $\ln n$ уровней, а отношение размерностей пространств последовательных уровней равно e . Следовательно, применение такой иерархии может сократить ожидаемое время поиска с $O(n)$ до $O(\log n)$.

В реальных задачах построение оптимальной иерархии, вообще говоря, невозможно, тем не менее увеличение числа уровней в общем случае приводит к сокращению поиска. В частности, планировщик ABSTRIPS [48] на пяти задачах достиг примерно логарифмического уменьшения времени поиска решения по сравнению с планировщиком STRIPS.

Один из более простых вариантов иерархического планирования — откладывание выполнения некоторых предусловий состоит во введении некоторой иерархии условий. Условия на низших уровнях иерархии считаются менее важными по сравнению с условиями на высших уровнях, и их выполнение может быть отложено. План строится поэтапно, начиная с учета ограничений только высшего уровня, и на каждом этапе добавляются действия, направленные на выполнение очередных менее существенных условий. Множество правил при этом остается единым, какого-либо ранжирования правил в соответствии с иерархией не требуется. Такой вариант был реализован в системе ABSTRIPS [48].

В системе MOLGEN [49, 50] для детализации плана в процессе иерархического планирования вводятся дополнительные ограничения и план строится путем распространения этих ограничений по принципу наименьшего принуждения (см. ниже). Как

мы увидим в подразделе 4.4, ограничения как средство организации поиска широко используются в экспертных системах для календарного планирования.

В ранних иерархических планировщиках, например NOAN, ABSTRIPS, в случае неудачного поиска на низших уровнях возврат на более высокие уровни для выбора другого варианта плана был невозможен, в более поздних, начиная, по-видимому, с планировщика NONLIN [51], это становится возможным. Обсуждение различных вариантов организации поиска в иерархии пространств можно найти в [52, 58].

Иерархия целей и упорядочение целей. На процесс построения плана может существенно влиять очередность достижения целей. Это относится как к исходным целям (когда их несколько), так и к предусловиям действий, которые становятся подцелями при раскрытии вершин в дереве поиска. Неудачная очередность может резко увеличить объем поиска при построении приемлемого плана или даже не позволить вообще найти его. Поэтому более общий вариант иерархического планирования заключается во введении иерархии целей. На верхнем уровне могут находиться такие цели, которых труднее всего достигнуть, которые определяют существенные свойства пространства поиска или на которые не влияют любые планируемые действия. Однако и при наличии общей иерархии целей в пределах каждого уровня могут потребоваться изменения очередности целей для обеспечения полиоты поиска.

Принцип наименьшего принуждения (наименьших свершений). Этот принцип управления поиском состоит в том, что если в данный момент для выбора решения в некоторой подзадаче не хватает нужной информации, то оно откладывается до тех пор, пока не появится достаточно информации для принятия «обоснованного» решения. Принцип наименьшего принуждения позволяет скоординировать процесс поиска решения с наличием необходимой информации. При этом возможны и тупиковые ситуации, когда процесс поиска останавливается, так как ни в одной из подзадач нет достаточных оснований для принятия решения. В системе MOLGEN в этих случаях включается эвристическая процедура, осуществляющая «угадывание».

Линейность (нелинейность). При наличии нескольких целей планирования с точки зрения способа достижения этих целей все планировщики можно разбить на две группы: линейные и нелинейные. Более простые линейные планировщики, начиная с системы STRIPS, опирались на «предположение линейности», состоящее в следующем: отдельные цели между собой не связаны (или связаны очень слабо); поэтому можно построить план для достижения первой цели, дополнить его действиями, обеспечивающими достижение второй цели, и т. д. В нелинейных планировщиках, первым из которых был планировщик

NOAH [53], план представляется не в виде строгой последовательности действий, а в виде частично упорядоченной сети действий и целей, и отношения порядка между ними вводятся лишь по мере необходимости в ходе решения задачи.

И в линейных, и в нелинейных планировщиках в процессе достижения разных целей возможно возникновение конфликтов. Кроме планировщика STRIPS, где такие конфликты просто игнорировались, во всех последующих планировщиках уже были предусмотрены различные способы их обнаружения и устранения [44].

5. Приспосабливающее планирование (opportunistic planning) [54, 78]. Этот принцип управления поиском, в отличие от иерархического, является двунаправленным (от целей к данным и от данных к целям) и позволяет составлять план из отдельных частей (которые могут строиться независимо, возможно, на разных уровнях абстракции и по разным моделям), выбирая наиболее подходящую из них. Он обеспечивает взаимодействие нескольких независимых источников знаний, т. е. процессов решения задачи, с помощью структурированной глобальной базы данных, «классной доски» (blackboard) [95].

Представления о ресурсах и времени. Сущностные особенности задач календарного планирования — использование ограниченных ресурсов, в т. ч. времени, параллельное выполнение нескольких операций, наличие неопределенности при исполнении расписания в реальных производственных условиях и др. требуют присутствия в планировщике развитых средств для оперирования такими понятиями, как ресурсы, временные ограничения, внешние события. Первыми планировщиками, обладающими такого рода возможностями, хотя и ограниченными, были системы NONLIN [51], SIPE [55] и DEVISER [56]. Так, в планировщике NONLIN на основе информации о длительности отдельных действий вычисляется самое раннее и самое позднее возможное время начала каждого действия и могут определяться критические пути выполнения действий. Допускается использование нескольких ограниченных ресурсов, предельные объемы потребляемых ресурсов используются в процессе поиска для сравнения различных альтернатив и выбора варианта с меньшим потреблением ресурсов.

Система SIPE допускает наличие конкурирующих запросов на использование разделяемых объектов, рассматриваемых как ресурсы, в т. ч. потребляемые. Однако SIPE обрабатывает только взаимодействия типа «действие a_1 и a_2 требуют одного и того же ресурса». Никакого общего представления о взаимодействиях параллельных действий здесь нет.

Планировщик DEVISER обладает более развитыми возможностями во временной области: вводится абсолютная шкала времени, позволяющая задавать интервалы времени (временные окна) для осуществления любых действий и целей, можно

описывать внешние события, задавая их воздействия в определенный момент времени. В процессе планирования анализируются соотношения между временными окнами отдельных действий, и под влиянием ограничений, налагаемых другими действиями, эти окна постепенно сужаются. В то же время DEVISER не дает средств для более общего представления, допускающего относительное временное упорядочение: нельзя выразить требование, чтобы два действия не перекрывались во времени, нельзя представить, что внешнее действие произойдет в произвольный момент времени (например, между 2.00 и 2.15) — должны быть заданы точные моменты начала и окончания. В планировщике O-Plan, являющемся дальнейшим развитием планировщика NONLIN, принято единообразное представление временных ограничений и периодов использования ресурсов в виде пары чисел (минимальное, максимальное), которая связывает фактические моменты начала и окончания действий, длительность действий и задержки в выполнении действий.

Значительно более гибкие возможности оперирования со временем предоставляют некоторые временные логики, подробнее о которых мы скажем в подразделе 4.3.

Условные и итеративные конструкции. Большая часть методов управления поиском и способов разрешения конфликтов в автоматических планировщиках ориентирована на построение планов, представляющих собой полностью или частично определенные последовательности элементарных действий. Однако предпринимались и попытки генерирования планов, имеющих более сложную структуру, в частности содержащих условные конструкции вида IF... THEN... ELSE и итеративные конструкции вида REPEAT... UNTIL. Так, планировщик NOAH [53] мог пользоваться такими конструкциями в явной форме, в планировщике SIPE [55] могли выполняться итерации, которые повторяли запланированную последовательность действий с каждым элементом некоторого множества параметров.

Автоматические планировщики разрабатывались как исследовательские программы, ни один из них не стал промышленной системой, хотя некоторые убедительно демонстрировали свои возможности по решению реальных задач, например DEVISER [56] при планировании графиков космических полетов, ISIS [57] при оперативном планировании и управлении на цеховом уровне, MOLGEN [49, 50] — при планировании экспериментов по молекулярной генетике.

По мере развития общих методов решения задач, в т. ч. и классических задач планирования, приходило понимание того, что универсальные методы поиска, взятые сами по себе, оказываются, как правило, недостаточны для решения большинства реальных задач, что конкретные знания о данной предметной области и способах решения задач в этой области должны составлять часть той информации, на основе которой решается

задача. Взаимосвязь между поиском и знанием, составляющая один из фундаментальных принципов ИИ, может быть сформулирована следующим образом: добавление знаний способствует сокращению поиска, но если знания отсутствуют, то поиск является достаточным (хотя и дорогим) средством решения проблем. Важность использования конкретных знаний стала очевидной с появлением в конце 70-х годов экспертных систем, основанных на правилах.

4.3. Принципы построения экспертных систем и представления знаний

Наблюдаемый в последние годы огромный интерес к экспертным системам (ЭС) можно объяснить прежде всего тем, что ЭС ориентированы на решение широкого круга неформализованных или слабо формализованных задач, для которых применение ЭВМ считалось ранее мало реальным. ЭС дают при этом вполне приемлемые для практики решения, а пользоваться ими могут специалисты, не умеющие программировать. Вопросам методологии, разработки и применения ЭС посвящена обширная литература (см., например, монографии [58—60]). Тем не менее, учитывая относительную новизну этого направления, мы очень кратко рассмотрим, что же такое ЭС и каковы средства представления знаний в ЭС с учетом специфики их применения в области календарного планирования.

Начнем с пояснения нескольких понятий [58]. Под предметной областью понимается некоторая область окружающего мира или совокупность знаний о ней (например, дискретные производственные процессы). Понятие проблемной области включает в себя предметную область и круг задач, решаемых в этой области. Данные в ней — это исходные, промежуточные или конечные данные, связанные с решаемой в данный момент задачей, а знания — это любая информация, в т. ч. и конкретные факты, хранящаяся в ЭС независимо от того, решается в ней задача или нет.

Типичная ЭС содержит следующие компоненты: 1) база знаний, содержащая знания о проблемной области, например в виде правил; 2) рабочая память, хранящая данные (база данных); 3) механизм вывода (интерпретатор), находящий решения заданных прикладных задач; 4) подсистема взаимодействия с пользователем (экспертом) на естественном для него языке; 5) компонента приобретения знаний; 6) объяснительная компонента, поясняющая пользователю действия ЭС. Схема обобщенной ЭС приведена на рис. 2.

На стадии разработки ЭС эксперт в данной проблемной области с помощью инженера по знаниям наполняет систему знаниями. Объединение новых знаний с уже имеющимися в базе производит компонента приобретения знаний. Следует отметить, что именно приобретение знаний вызывает наибольшие

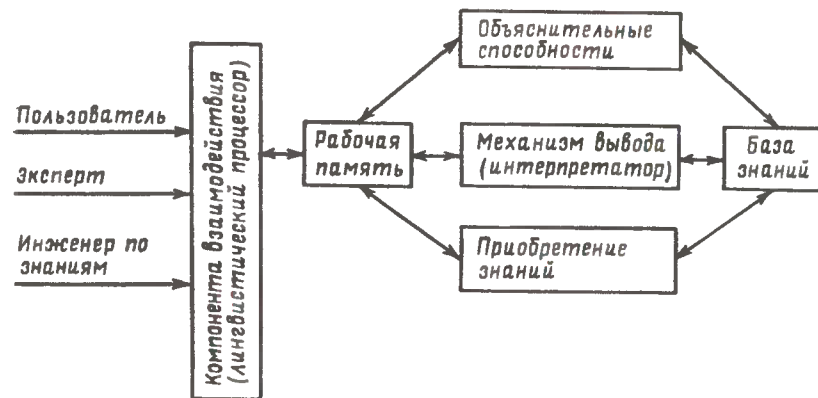


Рис. 2. Схема обобщенной ЭС

трудности в ходе создания ЭС и в значительной степени определяет удачный или неудачный исход ее разработки. В ходе решения задач с ЭС общается пользователь, для которого они решаются. Данные решаемой задачи после обработки подсистемой взаимодействия (она преобразует входные данные на ограниченном естественном языке во внутреннее представление ЭС) поступают в рабочую память. Интерпретатор на основе этих данных, имеющихся в ЭС знаний, и общих фактов о проблемной области решает задачу. Если же рассуждения или ответ системы не понятны пользователю, то объяснительная компонента может пояснить ход решения задачи.

В типичной ЭС можно выделить три уровня знаний: данные конкретных задач, знания о предметной области и стратегии управления решением задач. Рассмотрим средства представления знаний на первых двух уровнях, т. е. знаний о проблемной области. Здесь существуют четыре основных подхода: 1) логические модели, 2) продукционные системы и модели, управляемые образцами, 3) семантические сети, 4) фреймтовые модели. В отличие от первого, формального подхода последние три являются эвристическими.

Логические модели. Логические модели основаны на использовании некоторой формальной логической системы, чаще всего исчисления предикатов первого порядка. Достоинства логических моделей — это точность, корректность выводимых заключений, наличие единообразной формальной процедуры доказательства теорем. Основными недостатками являются комбинаторно возрастающая сложность, трудность использования при доказательстве эвристик, отражающих специфику конкретной проблемной области, а также отсутствие средств для структурирования информации. На формальной логике предикатов базируется язык программирования ПРОЛОГ — один из двух наиболее распространенных языков ИИ [93].

В автоматических планировщиках, рассмотренных в подразделе 4.2, использовались преимущественно логические модели. Там же были отмечены ограничения, накладывающие логические модели на основе стандартного исчисления предикатов первого порядка на представление знаний о планировании. В связи с этим были разработаны более сложные логические модели на основе различных временных или модальных логик [61—63]. Появляется общее понятие времени, не зависящего от действий исполнителя. Так, в [62] предложены логика временных интервалов, которая строится как логика 1-го порядка и устанавливает 13 возможных соотношений между двумя временными интервалами, а также метод вывода в этой логике, основанный на распространении ограничений. Эта логика имеет ряд программных реализаций на языке ЛИСП. Временная логика [63] также использует понятие интервала, хотя и в иной интерпретации.

На базе временной логики [62] в [64] разработана общая теория действий и времени, где с каждым интервалом ассоциируется один из трех объектов: свойство, событие или процесс. Свойство — это то, что «статически», т. е. в течение всего интервала истинно или ложно. Напротив, событие происходит на интервале, но не на каком-либо его подинтервале. Процесс представляет собой комбинацию понятий свойства и события.

Обе логики [62, 63] применимы для представления внешних событий, но в них нельзя представить, что может и чего не может сделать исполнитель плана. Поэтому в [61] логика интервалов [62] расширяется двумя модальностями: НЕИЗБ — для выражения простых возможностей во времени и ЕСЛИ ПОПЫТАТЬСЯ — для выражения сослагательных утверждений относительно действий. Утверждение (НЕИЗБ iP), где i — момент окончания интервала i , а P — некоторое утверждение, позволяет делать высказывания о том, что неизбежно на момент времени i , что возможно на момент i и что действительно истинно. Утверждение (ЕСЛИ ПОПЫТАТЬСЯ piP), где pi — некоторый пункт плана, позволяет делать высказывания о том, к чему привело бы исполнение пункта pi , будь он выполнен действительно, независимо от того, что истинно на самом деле.

Суперпозицией утверждений ЕСЛИ ПОПЫТАТЬСЯ можно представить взаимодействие совмещенных по времени пунктов плана и формальным способом определить, могут ли они быть выполнены одновременно и если да, то при каких условиях.

В логических моделях представления знаний отношения, существующие между отдельными единицами знаний, выражаются только с помощью тех небогатых средств, которые допускаются синтаксическими правилами используемой формальной системы. В отличие от формальных, эвристические модели имеют разнообразный набор средств для отображения специфики той или иной предметной области. Поэтому эвристические моде-

ли в целом превосходят логические как по возможностям адекватного представления различных проблемных областей, так и по эффективности используемых правил вывода.

Продукционные системы. Наиболее часто применяемая модель представления знаний в ЭС — это продукционные системы (ПС). Они берут свое начало с работ Поста, который в 1943 г. ввел термины продукция и каноническая (продукционная) система, но отличаются от формальных ПС, во-первых, использованием более разнообразных правил, и, во-вторых, включением эвристической информации о конкретных проблемных областях.

Основными элементами ПС являются данные, хранимые в рабочей памяти (или глобальной базе данных), множество правил продукции и система управления. Данные доступны для всех правил, связь между правилами осуществляется только через данные. Правило продукции в общем случае имеет структуру вида

ЕСЛИ<предусловие>ТО<действие>,

где предусловие — это логическая комбинация утверждений о состоянии данных, а действие — некоторая операция модификации данных. Сложность действий может изменяться в значительных пределах — от простой операции присваивания до функций любой степени сложности. При использовании некоторого правила проверяется выполнимость предусловия для данных в рабочей памяти, и если оно удовлетворяется, то данное правило является применимым. Система управления выбирает, какое именно из применимых правил следует фактически использовать (применение выбранного правила модифицирует рабочую память), и продолжает процесс проверки и применения правил до тех пор, пока рабочая память не будет удовлетворять некоторому правилу остановки (терминальному условию).

ПС делятся на ПС с прямым выводом, т. е. управляемые данными (предусловиями правил), и ПС с обратным выводом, т. е. управляемые целями (действиями правил). Традиционно под ПС принято понимать ПС с прямым выводом.

Представление знаний с помощью ПС обладает следующими достоинствами: 1) модульность организации знаний, 2) независимость правил, выражающих самостоятельные фрагменты знаний, 3) удобство модификации знаний, 4) отделение управляющих знаний от предметных, что позволяет применять различные стратегии управления. Кроме того, как показали психологические исследования, человек, рассуждая, пользуется правилами, подобными продукционным, т. е. правилами вида «условие — действие». Основные недостатки ПС заключаются в более низкой эффективности по сравнению с методами традиционного программирования и в трудности отображения иерархии знаний и потока управления.

ПС составляют частный случай более общей модели представления знаний — т. н. систем модулей, управляемых образцами [42, 58]. Различие между ними состоит прежде всего в сложности используемых модулей: если в ПС модулями являются отдельные правила продукции, смысл которых понятен пользователю (эксперту), незнакомому с программированием, то в качестве модулей в общем случае используются сложные автономные фрагменты знаний, представленные программами, смысл которых непрограммисту уже непонятен.

Семантические сети. В основе семантических моделей лежит понятие сети, образованной помеченными вершинами и дугами. Вершины представляют некоторые сущности (объекты, события, процессы), а дуги — отношения между соответствующими сущностями. Семантическая сеть называется простой, если вершины не имеют собственной внутренней структуры, и иерархической в противном случае. Семантические сети используются преимущественно в таких предметных областях, где существенную роль играют причинно-следственные связи, например в медицинской диагностике. К числу достоинств семантических сетей следует отнести гибкость, удобство отображения иерархии и прослеживания ассоциаций, а к недостаткам — трудоемкость обработки исключений и возможная неоднозначность сущностей, приписанных вершинам. Представление знаний семантическими сетями применяется, в частности, в известных ЭС PROSPECTOR для геологической разведки и INTERNIST для медицинской диагностики и терапии.

Фреймовые модели. Стремление соединить в рамках одной модели достоинства различных подходов к представлению знаний привело к созданию фрейм-представления или фреймовой модели [65]. Фрейм — это сложная структура данных, предназначенная для представления некоторого объекта или некоторой стандартной ситуации. Фрейм состоит из частей — слотов, соответствующих свойствам объекта или объектам, составляющим данную ситуацию, и отношений между слотами. С каждым слотом фрейма связаны описания условий, которые должны быть соблюдены, чтобы могло произойти означивание слота. Обычно различают фреймы-прототипы и фреймы-примеры. Первые представляют собой собственно знание: в качестве значений слота они содержат либо некоторый текст — общий для всех фреймов-примеров, либо некоторые правила и ограничения, связанные с заполнением значений слотов во фреймах-примерах, порождаемых данным фреймом-прототипом. Система работает с фреймами-примерами, которые получают из соответствующих фреймов-прототипов путем полного или частичного означивания слотов. Родственные фреймы могут быть связаны в систему фреймов, содержащую описания причинных, временных и др. зависимостей между этими фреймами. Если воспользоваться аналогией с описанием процедур в языках программирования,

то фрейм-прототип соответствует описанию процедуры, а фрейм-пример соответствует вызову процедуры. Отличаются же фреймы от описаний процедур тем, что: а) фреймы могут вызываться не только по имени, но и по соответствию текущей ситуации и ситуации, описываемой этим фреймом; б) фрейм-представление описывает ситуацию в семантических, а не в синтаксических терминах и метатерминах. Мощные выразительные возможности фрейм-представления обусловлены наличием встроенных механизмов, поддерживающих такие свойства, как наследование, вызываемые процедуры (демоны), умолчания. На основе концепции фреймов разработан ряд конкретных языков представлений знаний — KRL, FRL, SRL и др. В частности, язык SRL был использован в ряде проектов по разработке прикладных систем ИИ в области организационного управления, в т. ч. в известной ЭС календарного планирования на цеховом уровне ISIS [57] (см. подраздел 4.4). В языке SRL (Schema Representation Language) [66] реализованы наряду с указанными выше концепциями и ряд новых возможностей. Среди них — возможность задания отношения наследования самим пользователем (в дополнение к уже имеющимся в языке отношениям is-a и instance), при этом наследуются только слоты и значения по усмотрению пользователя. Некоторые примеры применения средств представления знаний в ЭС для календарного планирования будут приведены ниже.

4.4. Реализация экспертных систем для календарного планирования

В настоящее время в ряде стран, прежде всего в США, Великобритании, Франции, Японии, разработаны или находятся в стадии разработки тысячи ЭС для самых разных приложений. Для облегчения и ускорения создания прикладных ЭС имеется большое число специальных программных средств, называемых инструментальными системами или ЭС-оболочками, а также специализированные ЭВМ: ЛИСП-машины и ПРОЛОГ-машины. Обзор современного состояния в этой области можно найти в [60, 67].

К сожалению, пока лишь небольшое число ЭС достигло стадии промышленной или коммерческой системы, преобладающая же часть находится на стадии прототипов. Наибольшее число ЭС, в основном продукционного типа, разрабатывается для естественно-научных областей, например медицины, химии, геологии. Устойчивость и возможность накопления знаний в этих областях создают предпосылки для тиражирования знаний и внушают к ЭС доверие со стороны пользователей.

По сравнению с этими традиционными областями применение ЭС в организационном управлении — направление сравнительно новое и менее развитое. Одна из основных причин этого обусловлена, по-видимому, проблемой приобретения знаний

[58—60]: эксперты часто испытывают затруднения при описании предметной области и формулировании правил, предложенные экспертом правила бывают неоднозначны, не свободны от противоречий и не устойчивы во времени, а также могут существенно отличаться от правил, предложенных другими экспертами. В сфере управления производством, одним из элементов которого является календарное планирование, эти трудности усугубляются тем, что производственные системы, как правило, функционируют в конкретных условиях и подвержены разнообразным внешним воздействиям.

Дать объективный детальный анализ актуального состояния в области создания ЭС для календарного планирования весьма затруднительно. С одной стороны, имеющиеся в литературе данные свидетельствуют о разработке или существовании в виде прототипов десятков таких ЭС, существенно различающихся между собой, и поток подобной информации растет. С другой стороны, эта информация, часто неполная, носящая во многих случаях рекламный или коммерческий характер, не дает достаточно полного и достоверного представления об отдельных ЭС. Большие трудности вызывают также вопросы оценки реальных возможностей и качества решений конкретных ЭС и их сравнения между собой. Во-первых, потому что эти ЭС предназначены для решения разных задач в различных условиях, во-вторых, как правило, они являются прототипами и находятся в состоянии перманентной модификации, в-третьих, до сих пор отсутствуют общепринятые подходы и процедуры оценки качества ЭС [59, 60]. Поэтому мы попытаемся дать лишь общую характеристику состояния разработок ЭС в данной предметной области и приведем ряд примеров реализации конкретных систем.

Классификация систем. С методологической точки зрения ЭС для календарного планирования можно разделить на две группы:

— ЭС в строгом смысле, решающие задачи только на основе экспертных знаний о конкретной проблемной области,

— гибридные системы, использующие совместно экспертные знания и алгоритмические методы исследования операций.

Среди гибридных систем можно выделить методо-ориентированные ЭС, в которых доминируют алгоритмические знания, а экспертные знания служат лишь средством помощи пользователю в их применении. Некоторые гибридные системы, содержащие большое число методов и обладающие основанной на знаниях способностью к «пониманию» решаемой задачи и выбору наиболее подходящих методов, приближаются по своему характеру к основанным на знаниях системам поддержки принятия решений, разрабатываемым в области организационного управления [68].

Виды решаемых задач. В силу характера рассматриваемого подхода и ориентации на реальные приложения задачи, решаемые

ЭС, как правило, не имеют математической формулировки и систематизировать их в соответствии с формальной классификацией, например принятой в теории расписаний (см. раздел 1), обычно не удается. В целом можно сказать, что ЭС ориентированы преимущественно на те категории задач, которые не поддаются или с трудом поддаются решению при помощи традиционных подходов, рассмотренных выше.

Среди этих задач преобладают задачи оптимизационного характера, хотя вопрос о получении оптимальных или близких к оптимальным расписаний, как правило, не рассматривается. В большинстве случаев оценка качества генерируемых ЭС расписаний сводится к утверждениям о том, что они «не уступают» расписаниям, получаемым вручную, «сравнимы» с ними и т. д. Заметное место занимают и задачи типа удовлетворения ограничений [42]. Учитывая сказанное, мы перечислим основные виды задач с точки зрения приложений.

1. Задачи календарного планирования производства. Эти задачи имеют свою специфику и решаются по-разному для традиционных производственных систем и гибких производственных систем (ГПС). Применительно к традиционным системам ЭС решают как статические, так и динамические задачи, например ISIS [57], SOJA [69], системы, описанные в [86, 85, 70, 71]. Как известно, в реальных производственных условиях основную трудность составляет не столько планирование, сколько многократное перепланирование, поэтому ряд ЭС ориентирован именно на оперативную модификацию расписаний на стадии их исполнения [70, 71]. Большая же часть исследований посвящена разработке основанных на знаниях методов и ЭС для составления расписаний в ГПС. Присущие ГПС особенности [72, 73] существенно изменяют и усложняют эти задачи по сравнению с традиционными производственными системами и, как правило, требуют построения динамических расписаний в рамках интегрированных систем планирования и управления ГПС в реальном времени. Соответственно и спектр разрабатываемых здесь подходов и систем чрезвычайно широк — от ЭС с традиционной архитектурой и стандартными средствами представления знаний до гибридных систем, имеющих самую разнообразную архитектуру и сочетающих разные методы представления знаний и решения задач (см., например, [72, 74, 75, 76, 77, 78]). Детальное рассмотрение этих вопросов заслуживает самостоятельного обзора, поэтому далее мы не будем касаться ЭС, ориентированных на работу в ГПС.

2. Выбор маршрутов и составление графиков движения транспортных средств, например [79—81].

3. Прочие задачи (составление расписаний учебных занятий [90], распределение шлюзов аэропорта между рейсами самолетов [82], комплектование и составление графиков работы научной аппаратуры на борту космических объектов [83, 84]).

Представление знаний. Как правило, применяются эвристические, а не логические модели. Нам известны лишь два случая использования временных логик, в т. ч. [71]. Широко применяются ПС, причем ЭС, решающие относительно простые задачи, строятся только на основе ПС [79, 82]. В более сложных ЭС используются фреймовые модели [69, 83, 57] или смешанные представления. Например, часто встречается сочетание фреймовой модели для описания предметной области и правил продукции для представления операционных знаний [80, 85]. Судя по литературе, не находят применения семантические сети.

Стратегии управления решением задач. В более простых ЭС производственного типа применяется обычно процедура прямого вывода на одном уровне [79, 82]. В ЭС для решения более сложных задач предусмотрены иерархические процедуры поиска, содержащие, например, 4 уровня [57] или 5 уровней [86]. Широкое распространение получил метод поиска, направляемого ограничениями (ПНО), разработанный специально для решения задач календарного планирования [57] (мы его рассмотрим более подробно при описании системы ISIS). Процедура ПНО применяется также в системе SOJA [69], системах, рассмотренных в [62, 86].

После такой общей характеристики рассмотрим кратко несколько конкретных систем, которые, судя по доступной информации, представляют интерес, останавливаясь преимущественно на методической стороне этих систем. Данные о практической реализации этих систем, а также основные сведения о ряде других систем включены в приводимую далее таблицу, содержащую сводку основных сведений о ряде ЭС для календарного планирования (без ЭС для ГПС).

Характерным и наиболее известным представителем ЭС первой группы является система ISIS [57]. Это была, по-видимому, вообще первая система ИИ, ориентированная на задачи календарного планирования. Она разрабатывалась в университете Карнеги-Меллона (США) применительно к условиям планирования производства на цеховом уровне и апробировалась на заводе по производству турбинных лопаток фирмы Westinghouse. Развитый здесь подход к решению задач календарного планирования оказал заметное влияние на многие последующие разработки ЭС в этой области.

Согласно данному подходу, построение расписания трактуется как процедура ПНО, причем вследствие конфликтного характера многих ограничений в производственной системе одного лишь метода распространения ограничений недостаточно (как в обычных задачах удовлетворения ограничений [42]) и для получения приемлемого решения приходится смягчать некоторые ограничения и искать решения, наилучшим образом удовлетворяющие всем ограничениям.

В результате анализа ограничений задач календарного планирования и их влияния на процедуру построения расписания было выделено пять категорий ограничений: 1) организационные (директивные сроки, объемы незавершенного производства, затраты и т. д.), 2) физические (времена переналадки и обработки, физические ограничения машин и др.), 3) причинные (очередность операций, потребности в ресурсах и т. п.), 4) доступность ресурсов с учетом плановых и неплановых простоев машин и др. факторов, 5) предпочтения, отражающие мнение персонала (например, о желательности или нежелательности замены одной машины другой).

Доминирующая роль ограничений в процессе построения плана на основе ПНО требует адекватного представления знаний об ограничениях. В системе ISIS знания о каждом ограничении отражают следующие аспекты:

- возможности релаксации — указание допустимых вариантов данного ограничения и предпочтений между ними,
- важность — относительная значимость в процессе планирования,
- релевантность — условия, при которых ограничение должно использоваться,
- взаимосвязи — связь данного ограничения с другими ограничениями,
- способ генерации — описание механизма формирования и распространения ограничения.

Могут использоваться практически любые ограничения, задаваемые пользователем. Примером представления ограничений средствами применяемого в системе ISIS фреймоподобного языка SRL [35] может служить рис. 3. Данное ограничение задает диапазон значений, которые может иметь слот ДИРЕКТИВНЫЙ СРОК, связанный со схемой на языке SRL, описывающей некоторый ЗАКАЗ. Набор допустимых значений (т. е. релаксация) для этого ограничения задается отдельной схемой (рис. 4). Принимается, что эта релаксация непрерывного типа, она ограничивает значения слота областью календарных дат и задает в этой области кусочно-линейную функцию полезности, позволяющую путем интерполяции оценить приемлемость каждого выбираемого значения.

Построение расписания представляет собой многоуровневую иерархическую процедуру ПНО в пространстве всех возможных расписаний. Поиск ведется сверху-вниз, связь между уровнями осуществляется посредством обмена ограничениями.

На каждом уровне выполняются три фазы: 1) предварительный анализ (определение границ пространства поиска на данном уровне), 2) поиск решения в данном пространстве, 3) пост-анализ (оценка качества найденного решения). Если качество признается приемлемым, то результат преобразуется в форму ограничений и передается на очередной нижележащий уровень,

ЭС для календарного планирования

Наименование системы и/или источник информации	Область применения	Средства представления знаний	Применяемые методы планирования и/или управления поиском (выводом)	Статия разработки; средства реализации	Режим решения задач
ISIS [57]	Календарное планирование на цеховом уровне	Фреймы (схемы)	Иерархическое планирование ПНО	Действующий прототип (1984); Язык SRL	Автоматический и диалоговый
SOJA [69]	Сменное (суточное) планирование дискретного производства	Правила и фреймы	ПНО, прямой и обратный вывод	Прототип (1985); Язык Franz LISP; VAX 11/780	Автоматический
Система [86]	То же	Фреймы, правила про-дукции	Иерархическое планирование, ПНО	Прототип (1986); Язык Virginia Tech Prolog, ЭС-оболочка GUESS-1; VAX-11/780	»
Система [80]	Выбор маршрутов и составление графиков движения автотранспорта	То же	Прямой вывод	Прототип (1987); Knowledge Craft; Язык SRL	Диалоговый
Система [79]	То же	Правила про-дукции	Прямой вывод	Прототип (1988); Язык ТУРБО-ПРОЛОГ; ЭВМ типа IBM	Диалоговый
GATES [82]	Составление графика рас-пределения шлюзов между рейсами в аэропорту	То же	Распространение ограничений	РС/XT	»
Система [90]	Составление расписаний учебных занятий	»	ПНО	Действующий прототип (1988); Язык ПРО-ЛОГ (~1000 строк); Персональная ЭВМ PARLOG; персональ-ная ЭВМ (МП 8086, ОЗУ 640 Кбайт)	Автоматический
MAN—SCHED [85]	Календарное планирование ремонтов оборудования в энергосистемах	Фреймы, пра-вила про-дукции	Метод ветвей и границ, прямой вывод	Прототип (1987)	Диалоговый
Динамический перелани-ровщик [71]	Оперативная корректировка производственных планов (графиков)	Правила про-дукции, ло-гика времен-ных интер-валов	ПНО	Прототип (1988); Си-стема ART; ЭВМ Symbolics и DEC	Диалоговый
Система [81]	Составление расписаний движения поездов	Смешанное представление	Несколько мето-дов	Прототип (1988); Систе-ма EURIKA II; рабо-чая станция Hitachi ES 330	»
SCHEDULE [89]	Решение задач теории рас-писаний	Алгоритмы, пра-вила, в ви-де графов	Алгоритмы теории расписаний	Начальный вариант (1988); Язык Паскаль; ЭВМ типа HP 1000	»
Система [70]	Оперативная корректировка производственных планов линии по производству интегральных схем	Специальная модель, пра-вила про-дукции	Двухнаправленное распространение изменений	—	»
МОРА [83]	Поддержка пользователя при составлении графика работы научной аппара-туры на борту ИСЗ	Фреймы (схе-мы)	Иерархическое планирование	Прототип (1986); Систе-ма ART 2.0+Zeta LISP; ЭВМ Symbolics 3640	Диалоговый
EMPRESS [84]	Поддержка пользователя при планировании загруз-ки и составлении графика использования науч-ной аппаратуры на борту космического корабля Shuttle	Правила про-дукции, огра-ничения	Иерархическое планирование	Прототип (1986); Язык Zeta LISP; ЭВМ Sym-bolics 3600	»

Примечания. 1. Проверка означает отсутствие данных. 2. Стадия разработки указана, как правило, по данным на год публикации (в скобках). 3. Термин «диалоговый режим» может иметь здесь весьма широкий смысл — от прос-того инициирования пользователем очередных указаний в меню шагов решения до развитого диалога с пользователем, имеющим статус ЛПР.

```

{{ограничение директивный срок
 {is-a диапазон ограничение
  ВАЖНОСТЬ:
  КОНТЕКСТ: время
  ОБЛАСТЬ:
    диапазон: (тип is-a заказ)
  ОТНОШЕНИЕ: директивный срок
  ОГРАНИЧЕНИЯ:
    диапазон: (тип is-a допустимые значения)}}
ПРИОРИТЕТ: }}

```

Рис. 3. Схема «ограничение директивный срок»

```

{{релаксация директивный срок
 {is-a непрерывная релаксация
  ОБЛАСТЬ: календарные даты
  КУСОЧНО-ЛИНЕЙНАЯ ПОЛЕЗНОСТЬ:
  МЕТОД: интерполяция}}}

```

Рис. 4. Схема «релаксация директивный срок»

в противном случае производится возврат: пространство поиска на текущем или более высоком уровне модифицируется (путем релаксации одного или нескольких ограничений) с соответствующей передачей управления.

В системе ISIS принята 4-уровневая иерархия поиска: уровень 1 — выбор очередного заказа (с учетом приоритета и директивного срока), уровень 2 — планирование выполнения данного заказа по методу критического пути (его результатом являются самые ранние времена начала и самые поздние времена окончания всех операций по заказу), уровень 3 — детальное распределение ресурсов для варианта, найденного на уровне 2, что дает почти готовое расписание, уровень 4 — уточнение графиков использования ресурсов для данного расписания.

Система ISIS может работать как в автоматическом, так и в интерактивном режиме, при этом пользователь ведет построение расписания, его модификацию или ввод оперативной информации также в соответствии с принятой 4-уровневой моделью. Общение с пользователем осуществляется посредством меню, фраз ограниченного английского языка, средств машинной графики и вопросно-ответной подсистемы.

Система SOJA [69] предназначена для сменного планирования дискретного производства на цеховом уровне. Принятая процедура планирования состоит из двух этапов: 1) выбор пунктов плана, называемых далее фазами и подлежащих выполнению с учетом имеющихся заказов и состояния производства; 2) упорядочение этих фаз и составление графика работы каждой машины. Под фазой производства здесь понимается после-

довательность операций по обработке данной партии изделий, выполняемых на данной машине с одним и тем же набором инструментов. Построение графика в системе SOJA производится также на основе ПНО. Все учитываемые ограничения делятся на две группы: 1) «жесткие» ограничения технологического характера (очередность операций, правила использования ресурсов и др.), 2) «мягкие» ограничения (соблюдение директивных сроков, равномерность загрузки оборудования и т. п.). Эти ограничения преобразуются в правила, управляющие процессом удовлетворения ограничений первой группы.

Каждое ограничение записывается в виде дизъюнкции некоторых фактов. Элементарный шаг решения состоит в выборе и распространении одного из этих фактов. Распространение фактов производится с помощью т. н. графа расписания (ГР), вершинами которого являются выбранные на этапе 1 фазы (в т. ч. специальная вершина B , соответствующая началу смены), а дуги отображают временные соотношения между фазами. В системе SOJA выделяется шесть категорий ограничений:

1) самое раннее время начала каждой фазы F , определяемое временем доставки партии на эту машину и смены инструмента; такое ограничение содержит один факт и вносит в ГР одну дугу $B \rightarrow F$;

2) необходимость выполнения выбранной фазы F до конца дня (планового периода); здесь также один факт и одна дуга $F \rightarrow B$;

3) ограничение предшествования вида (F_1 предшествует F_2); в ГР ему соответствует одна дуга, вес которой зависит от ряда параметров (используемые для выполнения фаз F_1 и F_2 машины, времена переналадки и т. д.);

4) разделение ресурсов: если фазы F_1 и F_2 используют один и тот же ресурс, то должно выполняться условие (F_1 предшествует F_2) ИЛИ (F_2 предшествует F_1);

5) ограничения на две последовательные фазы, относящиеся к одной и той же партии деталей и выполняемые на разных машинах M_1 и M_2 ;

6) ограничения, связанные с резервированием мест для хранения деталей рядом с машинами

В процессе поиска расписания для выбора одного из фактов в дизъюнктивных ограничениях используются приоритетные правила, которые позволяют, например, выделить наиболее важные категории ограничений, отсрочить рассмотрение некоторых ограничений, вызвать некоторую эвристическую функцию, чтобы выбрать для продолжения наиболее «подходящий» факт и др. Используемые приоритетные правила включают в себя правила, известные в теории расписаний, и правила, сформулированные с помощью работников цеха, для условий которого разрабатывался прототип.

Для представления знаний, в т. ч. правил, в системе SOJA применена фреймовая модель, погруженная в среду языка ЛИСП. Механизм вывода реализует как прямой, так и обратный вывод. Соответственно все правила делятся на три группы: прямые, используемые при прямом выводе для модификации текущего состояния данных; обратные, используемые при обратном выводе для доказательства положительных термов, входящих в условия других термов; правила-демоны, активизация которых наступает только вследствие применения других правил.

Прототип системы SOJA реализован применительно к условиям цеха стального листа фирмы Alsthom-Atlantique (Франция). В экспериментах на условных данных составление графика для 18 машин и 42 фаз производства на период 4 ч заняло 5 мин, а для 18 машин и 83 фаз на период 8 ч — 20 мин. Достигаемый средний коэффициент загрузки оборудования составил ~70%.

В [86] описывается прототип ЭС для составления расписаний в производственной системе с параллельными процессорами при наличии ограничений, отражающих отношения предшествования, предпочтения и конечность промежуточных емкостей, причем некоторые ограничения, например задающие безопасные уровни запасов и допустимые последовательности выпуска продукции, зависимы от времени. По этой причине последовательная процедура назначения операций, принятая, например, в системе ISIS [57], оказалась бы здесь весьма неэффективной вследствие большого числа возвратов и желательно было бы рассматривать все процессоры и работы одновременно. Поэтому был принят комбинированный подход, сочетающий принцип иерархического планирования, ПНО и разбиение периода планирования на интервалы. Тестирование прототипа на реальных данных за 8 суток работы производственной системы подтвердило большую степень сходства расписаний, генерируемых ЭС, и расписаний, составленных персоналом вручную, с точки зрения соблюдения ограничений и величины отклонения в случае их нарушения.

В [80] описывается ЭС для задач маршрутизации и диспетчеризации движения автотранспорта. Решаемые задачи характеризуются следующими основными свойствами: целевая функция — минимизация транспортных расходов по удовлетворению детерминированного спроса в узлах транспортной сети, наличие разнородного парка автомашин и нескольких пунктов парковки, необходимость соблюдения «временных окон» при обслуживании, возможность частичной догрузки и разгрузки в пути следования. В ЭС используются различные методы представления знаний: экспертные знания о правилах решения задач заданы в виде продукционных правил, понятия предметной области описываются фреймами (схемами языка CRL), для интерпрета-

тора правил совокупность этих фреймов составляет рабочую память.

Доводка прототипа до стадии промышленной системы производится у ее пользователя — одного из подразделений фирмы Digital Equipment Corp. (США). Генерируемые ЭС расписания сравнимы с расписаниями, составленными вручную. На реальных данных за 15 дней эксплуатации ЭС генерировала расписания, позволяющие сократить затраты на 2%.

Близкие задачи — выбор маршрутов и составление расписания движения школьных автобусов в сельской местности — составляют область приложения более простой продукционной ЭС из [79].

ЭС GATES [82] предназначена для составления суточного графика распределения шлюзов (выходов на посадку) между рейсами (прилетами и вылетами) самолетов компании TWA в аэропорту им. Кеннеди в Нью-Йорке и корректировки этих графиков в реальное время в случае каких-либо отклонений, например задержки прибытия самолета. Эта задача характеризуется наличием ряда недетерминированных ограничений и необходимостью учета предпочтений диспетчера, размерность задачи 100—115 рейсов и 30 шлюзов. ЭС решает эту задачу как задачу удовлетворения ограничений. Выделяются следующие пять категорий ограничений (в порядке убывания их важности): без исключений (например, физические размеры шлюзов), с исключениями (которые могут быть нарушены при определенных условиях), с изменяющимися временными допусками, связанные с интенсивностью движения по рулежным дорожкам и вызванные соображениями удобства.

ЭС GATES представляет собой продукционную систему, содержащую правила двух видов: разрешительные, описывающие условия допустимости назначения «шлюз А — рейс В», и конфликтные, выражающие недопустимость подобного назначения. Назначение шлюзов и рейсов производится в порядке убывания числа связанных с ними ограничений. При корректировке действующего графика ЭС получает текущую информацию о соблюдении расписания полетов из базы данных аэропорта. Описываемая версия ЭС позволяет скорректировать суточный график за ~30 с (вручную полная подготовка такого графика занимает 10—15 ч, а его ежесуточная корректировка ~1 ч).

Система для составления расписаний учебных занятий, описанная в [90], представляет интерес благодаря параллельной реализации на языке PARLOG. Данная задача, относящаяся также к задачам удовлетворения ограничений [42], описывается в виде немонотонной некоммукативной ПС. Для реализации немонотонного вывода используется очень упрощенный вариант системы поддержания истинности (Truth maintenance system). С точки зрения возможности распараллеливания при анализе ограничений все ограничения делятся на три типа: 1) многова-

риантные, генерирующие выходные сообщения (означенные значения) и требующие возврата в случае противоречия, 2) одновариантные, генерирующие выходные сообщения, 3) одновариантные без выходных сообщений. Система производит параллельный анализ ограничений типа 2 и 3. Однако эта классификация не является статической и существенно зависит от очередности анализа ограничений: при ее изменении меняется и число ограничений типа 1. Поэтому в целях повышения степени параллелизма и уменьшения числа возвратов выбирается такая очередность, при которой максимизируется число ограничений типа 2 и 3 и минимизируется число ограничений типа 1. Результаты тестирования на небольшом числе задач показали в несколько раз большее быстродействие этой системы по сравнению с обычной (последовательной) реализацией этой системы на языке ПРОЛОГ в тех же условиях.

Один из простейших вариантов гибридной системы календарного планирования — применение ЭС для решения отдельных подзадач в рамках общей схемы метода ветвей и границ. С точки зрения приведенной в [96] классификации гибридных методов, применяемых в дискретной оптимизации, это «использование эвристических приемов внутри точных схем». ЭС, позволяя находить разумные решения уже на ранней стадии поиска, способствует быстрому установлению более жестких границ и, следовательно, значительному сокращению поиска. С другой стороны, сохранение формальной процедуры управления поиском гарантирует получение хороших решений даже в случае слабой базы знаний.

Подобная схема применена в ЭС MAN-SCHED [85], предназначенной для планирования ремонтов оборудования в энергосистемах. В данной ЭС используется фреймовое представление знаний, причем имеется два вида фреймов: 1) статические, создаваемые первоначально при описании проблемной области и задающие два основных ее структурных элемента — «машины» как объекты обслуживания и временные интервалы; 2) динамические, описывающие промежуточные точки в процессе решения. Имеется две базы данных: в первой хранятся статические фреймы («машины» и «интервалы»), вторая база данных содержит фреймы, генерируемые в процессе построения расписания, и используется в качестве рабочей памяти.

Знания, характеризующие возможность или необходимость проведения ремонта конкретной машины в определенный интервал времени, представлены в виде производных правил, причем все множество правил относится ко всем машинам и всем интервалам и все правила независимы в том смысле, что консеквент (т. е. действие) одного правила не может быть пред-условием другого. Множество правил вызывается только по мере необходимости

Построение очередных допустимых решений по схеме ветвей и границ осуществляется сначала на основе экспертных правил, а в случае неудачи — с помощью стандартной процедуры. Целевая функция состоит в максимизации минимального резерва мощности энергосистемы на всех интервалах планового периода. Пользователь имеет возможность указать, какой из четырех возможных видов решений ему нужен (любое допустимое, допустимое с заданной оценкой отклонения от оптимального, одно оптимальное, все оптимальные), а также наблюдать на экране дисплея построение расписания интервал за интервалом.

Система MAN-SCHED была успешно применена для решения ряда известных из литературы задач составления графиков ремонта оборудования в энергосистемах, размерность самой большой из задач — 62 ремонта на 26 интервалах времени.

Динамический перепланировщик [71] представляет собой гибридную ЭС, помогающую управленческому персоналу быстро и компетентно реагировать на непредвиденные изменения в производственной обстановке, корректируя имеющийся план (график) в условиях дискретного серийного производства. Информация о текущем состоянии производства может поступать автоматически (например, от датчиков состояния оборудования) или вводиться вручную. Знания представлены в виде правил и объединяют как элементы алгоритмов планирования потребностей в материалах (системы типа MRP II) и известные эвристические правила диспетчеризации, так и собственно знания о целях производства и способах модификации расписаний. Все множество правил разделено на следующие группы: 1) операторы, которые на основе действующего плана и данных об изменениях ситуации генерируют различные альтернативные расписания, 2) правила вычисления фактов, например сроков выполнения заказов, 3) правила, описывающие ограничения (физические, ресурсные, предшествования и др.), 4) правила оценки и выбора наилучших альтернативных расписаний, 5) правила релаксации, смягчающие некоторые ограничения в случае необходимости.

Для представления знаний, связанных со временем, используется логика временных интервалов [52]. Построение альтернативных расписаний ведется на основе метода ПНО. Данная ЭС составляет основу для создания специализированных перепланировщиков для конкретных производственных условий.

Рядом фирм в районе г. Тулуза (Франция) совместно разрабатывается гибридная ЭС OPAL [87]. Теоретические знания здесь отделены от практических, причем и те и другие представлены в виде правил. Общая структура системы OPAL приведена на рис. 5.

База данных содержит описания производственной системы, решаемой задачи составления расписаний и текущего состояния процесса поиска. Теоретические знания содержатся в мо-

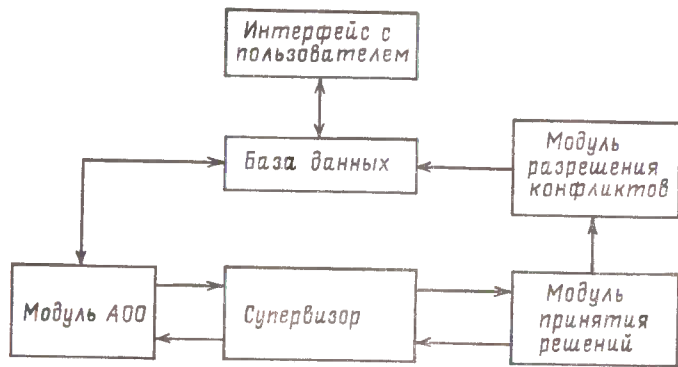


Рис. 5. Структура системы OPAL

дуле АОО (анализ на основе ограничений — constraint based analysis). Метод АОО [88] позволяет вывести ограничения на допустимую последовательность выполнения операций, исходя из неизменных свойств операций (длительность и набор ресурсов) путем анализа предельных сроков их проведения. Модуль принятия решения производит выбор отдельных операций в пределах ограничений, ранее заданных модулем АОО, в тех случаях, когда не все операции упорядочены и модуль АОО не способен более генерировать ограничения. Модуль принятия решений содержит знания практического характера — как известные эвристики, так и правила, отражающие конкретные технологические особенности и опыт управленческого персонала. Модуль разрешения конфликтов предназначен для преодоления ситуаций, связанных с отсутствием допустимых решений, путем ослабления отдельных ограничений или изменения распределения ресурсов. Супервизор управляет процессом поиска решения, включая поочередно модуль АОО (содержащий правила с более высоким приоритетом) и модуль принятия решений, а в случае, если этого недостаточно, вызывая модуль разрешения конфликтов.

Перспективный подход к созданию достаточно универсальных ЭС для календарного планирования рассматривается в [81]. В качестве прообраза такой ЭС разработан действующий прототип системы для составления расписаний движения поездов метрополитена. Основное содержание этого подхода — совместное применение нескольких способов представления знаний и развитых средств их обработки. Знания высокого уровня являются декларативными и задаются в виде иерархической сети «цели—стратегии», где каждая цель соединена со стратегиями, обеспечивающими ее достижение, а стратегия соединена со своими подцелями. Эти знания представляются сетью фреймов. Знания низкого уровня являются процедурными, обращение к ним производится стратегиями самого низкого уров-

ня. Эти знания представляются в основном в виде связанных с фреймами процедур, написанных на некотором эффективно исполняемом языке, например, СИ, а частично в виде продукционных правил. Общее расписание может составляться из нескольких частичных расписаний (например, для отдельных частей суток: часы пик, ночные часы и т. д.). Эксперты (процессы), генерирующие частичные расписания, моделируются совокупностью взаимодействующих объектов, т. н. акторов. При построении частичных расписаний акторы пользуются указанными выше знаниями, а с целью разрешения конфликтов и объединения частичных расписаний общаются между собой путем передачи сообщений. Использование акторами знаний в зависимости от конкретных ситуаций описывается продукционными правилами. Разработанная ЭС получила одобрение специалистов, занимающихся составлением расписаний движения поездов, намечено ее практическое использование.

Типичным примером методо-ориентированной ЭС является система SCHEDULE [89]. Авторы называют ее квазиэкспертной, рассматривая как первый шаг к созданию полноценной гибридной ЭС для календарного планирования. Она ориентирована на пользователя, обладающего некоторыми общими познаниями в области решения задач методами исследования операций, в частности умением описать реальную задачу с помощью стандартной модели.

Исходным для ЭС является описание задачи в рамках известной «классификационной формулы» задач теории расписаний [2] с помощью признаков $\alpha, \beta_1, \dots, \beta_5, \gamma$.

База знаний системы SCHEDULE состоит из базы правил, описывающих способы взаимного сведения разных задач, и базы алгоритмов. Для задания правил сведения задач каждому из семи параметров $\alpha, \beta_1, \dots, \beta_5, \gamma$ приписывается ориентированный граф сведения $D_i, i \in \{\alpha, \beta_1, \dots, \beta_5, \gamma\}$, вершины в котором суть возможные значения параметра i , и имеется дуга (a, b) , $b \neq a$, если и только если алгоритм, решающий задачу с параметром $i=b$, решает также задачу с параметром $i=a$. Представление правил сведения задач совокупностью графов $D_\alpha, \dots, D_\gamma$ оказывается удобнее, чем применение обычных продукционных правил. База алгоритмов содержит алгоритмы решения 37 различных типов задач теории расписаний вместе с оценками их временной сложности (по одному алгоритму для каждого типа задач, причем использованы только полиномиальные алгоритмы, даже если они дают только приближенные решения).

Поиск подходящего алгоритма для решаемой задачи типа $P' = \alpha' | \beta' | \gamma'$ осуществляет механизм вывода следующим образом. Пусть задаче P' в каждом из графов сведения $D_\alpha, \dots, D_\gamma$ соответствуют вершины $V_{\alpha'}, \dots, V_{\gamma'}$. Тогда задача типа P' считается сводимой к задаче типа $P'' = \alpha'' | \beta'' | \gamma''$, а P''

считается обобщением P' , что обозначается $P' \propto P''$, если в графе D_{α} существует путь из $V_{\alpha'}$ в $V_{\alpha''}$, в графе D_{β_1} путь из $V_{\beta_1'}$ в $V_{\beta_1''}$, ..., и в графе D_{γ} путь из $V_{\gamma'}$ в $V_{\gamma''}$ (вершины $V_{\alpha''}$, ..., $V_{\gamma''}$ в графах $D_{\alpha''}$, ..., $D_{\gamma''}$ соответствуют задаче P'').

В нетривиальном случае (когда подходящего для P' алгоритма в базе алгоритмов нет) пользователю выдаются все типы задач, такие, что $P' \propto P''$, и соответствующие алгоритмы, которые упорядочиваются в порядке неубывания сложности.

При отсутствии алгоритмов, пригодных для решения задачи P' или некоторого ее обобщения P'' , механизм вывода переходит к поиску алгоритмов, решающих некоторую задачу \bar{P} , являющуюся ослабленным вариантом задачи P' , т. е. $\bar{P} \propto P'$. Для пары задач \bar{P} и P' определяется величина $\rho(\bar{P}, P')$, называемая дальностью сведения и равная сумме длин семи кратчайших путей в графах D_{α} , ..., D_{γ} соответственно из вершины V_{α} в $V_{\alpha'}$, из V_{β_1} в $V_{\beta_1'}$, ..., из V_{γ} в $V_{\gamma'}$ (веса всех дуг при этом считаются равными 1). Для исходной задачи P' находятся такие релаксации $\bar{P} \propto P'$, для которых в базе алгоритмов имеются решающие их алгоритмы. Эти релаксации предлагаются пользователю в порядке неубывания дальностей $\rho(\bar{P}, P')$, а при их равенстве в порядке неубывания сложности соответствующих алгоритмов.

Хотя первый прототип системы SCHEDULE содержит все основные компоненты ЭС, некоторые ключевые вопросы решения пока очень упрощенно. В частности: 1) пользователь должен сам идентифицировать решаемую задачу, задав ее признаки, хотя и с помощью развитых средств поддержки диалога, 2) знания представлены в основном в алгоритмической форме в базе алгоритмов, а экспертные правила лишь помогают пользователю в выборе подходящего алгоритма, 3) объяснительные возможности системы очень бедны.

В дальнейшем в систему SCHEDULE предполагается внести ряд улучшений: во-первых, пополнить базу знаний правилами, позволяющими при выборе алгоритма решения учитывать не только тип задачи, но и другие ее исходные данные; во-вторых, единичные веса в графах сведения заменить весами, лучше отражающими «степень близости» соседних типов задач (принятые в качестве критерия выбора подходящей релаксации величины $\rho(\bar{P}, P')$ оказались неудачными) — определить такие веса можно было бы в процессе обучения системы; в-третьих, включить в базу алгоритмов больше интерактивных алгоритмов, так как вмешательство пользователя позволяет иногда находить допустимые решения исходной задачи даже при отсутствии для нее подходящего алгоритма

Три основных подхода к решению задач календарного планирования, рассмотренные в обзоре, имеют области применения использования не одного, а нескольких подходов, которые во многих случаях реальные задачи требуют для своего решения использования не одного, а нескольких подходов, которые применяются на разных этапах решения. С этой точки зрения весьма полезно, в частности, встраивать эвристические методы дискретной оптимизации в диалоговые процедуры решения задач календарного планирования и включать их в математическое обеспечение соответствующих ЭС.

Хотя рассмотренные методы являются сравнительно эффективными при решении достаточно широких подклассов задач календарного планирования, в дальнейшем целесообразно как расширение спектра используемых методов, так и повышение их эффективности. Представляется, что основные направления развития эвристических методов календарного планирования должны быть связаны с распараллеливанием алгоритмов и применением параллельных вычислительных систем (опыт параллельной реализации разнообразных алгоритмов дискретной оптимизации уже весьма значителен и положителен), совершенствованием диалогового взаимодействия пользователя с системой планирования, развитием средств приобретения экспертных знаний в ЭС и общим повышением «интеллектуального уровня» ЭС.

Следует отметить, что рассмотренные подходы к решению задач календарного планирования вполне применимы и к другим практически важным классам задач оптимизационного характера. В частности, речь может идти о задачах таксономии, задачах оптимального размещения и некоторых других. Разработка и детализация указанных подходов для соответствующих классов задач могла бы стать предметом дальнейших исследований.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ ИНФОРМАЦИИ

1. Теория расписаний и вычислительные машины / Под ред. Коффмана Э. Г.— М.: Наука, 1984.— 334 с.
2. Rinnooy Kan A. H. G. Machine Scheduling Problems: Classification Complexity and Computations.— Hague: Hijhoff, 1976.— 286 p.
3. Бельский А. С., Левнер Е. В. Применение моделей и методов теории расписаний в задачах оптимального планирования на грузовом транспорте // Автомат. и телемех.— 1988.— № 1.— С. 3—77.
4. Rodommer F. A., White K. F. A recent survey of production Scheduling // IEEE Trans. Syst. Man. and Cybern.— 1988.— 18, № 6.— P. 841—851.
5. Ben Arie D., Moodie C. L. Knowledge based routing and sequencing for discrete part production // J. Manuf. Syst.— 1988.— 6, № 4.— P. 287—297.
6. Zdrzalka S. Scheduling jobs on a single machine with periodic release data (deadline intervals) // Eur. J. Oper. Res.— 1989.— 40, № 2.— P. 243—251.

7. Конвей Р. В., Маквел В. Л., Миллер Л. В. Теория расписаний.— М.: Наука, 1975.— 236 с.
8. Танаев В. С., Шкурба В. В. Введение в теорию расписаний.— М.: Наука, 1975.— 356 с.
9. Рейнгольд Э., Нивельгель Ю., Део Н. Комбинаторные алгоритмы.— М.: Мир, 1980.— 324 с.
10. Корбут А. А., Финкельштейн Ю. Ю. Дискретное программирование.— М.: Наука, 1969.— 368 с.
11. Бурков В. Н., Рубинштейн М. И. Комбинаторное программирование.— М.: Знание, 1977.— 64 с.
12. Майника Э. Алгоритмы оптимизации на сетях и графах.— М.: Мир, 1981.— 323 с.
13. Харари Ф. Теория графов.— М.: Мир, 1973.— 300 с.
14. Оре О. Теория графов.— М.: Наука, 1980.— 336 с.
15. Авен О. И., Ловецкий С. Е., Моисеенко Г. Е. Оптимизация транспортных потоков.— М.: Наука, 1985.— 316 с.
16. Werra D. An introduction to timetabling // Eur. J. Oper. Res.— 1985.— 19.— P. 151—162.
17. White G. M., Wong S. K. S. Interactive timetabling in universities // Comput. Educ.— 1988.— 12, № 4.— P. 521—529.
18. Хедли Дж., Уайтин Т. Анализ систем управления запасами.— М.: Наука, 1969.— 512 с.
19. Кофман А., Крюон Р. Массовое обслуживание (теория и приложения) — М.: Мир, 1965.— 302 с.
20. Авен О. И., Гурин Н. Н., Коган Я. А. Оценка качества и оптимизация вычислительных систем.— М.: Наука, 1982.— 464 с.
21. Подчасова Т. П., Португал В. М., Татаров В. А., Шкурба В. В. Эвристические методы календарного планирования.— Киев.; Техника, 1980.— 126 с.
22. Moore M. An n job one machine sequencing algorithm for minimizing the number of late jobs // Manag. Sci.— 1969.— 15, № 1.— P. 117—125.
23. Simons B. B., Warmuth M. K. A fast algorithm for multiprocessor scheduling of unit-length jobs // SIAM J. Comput.— 1989.— 18, № 4.— P. 690—710.
24. Brucker P. J., Hamacher H. W. K-optimal solution sets for some polynomially solvable scheduling problems // Eur. J. of Oper. Res.— 1989.— 41.— P. 194—202.
25. Monma C. L., Rinnooy Kan A. H. A concise survey of efficiently solvable special cases of the permutation flowshop problem // Oper. Res.— 1983.— 17, № 2.— P. 105—119.
26. Рубинштейн М. И. Оптимальная группировка взаимосвязанных объектов.— М.: Наука, 1989.— 168 с.
27. Schultz C. R. An expediting heuristic for the shortest processing time dispatching rule // Int. J. Proc. Res.— 1989.— 27, № 1.— P. 43—52.
28. Рубинштейн М. И., Плитман А. Д. Комбинаторные методы группировки в задачах планирования и организации // Итоги науки и техники. Сер. Техн. киберн.— М.: ВИНТИИ.— 1986.— Т. 19.— С. 190—228.
29. Wee T. S., Magasine M. J. Assembly line balancing as generalized in packing // Oper. Res. Lett.— 1982.— P. 56—58.
30. Kampke T. Simulated annealing: use a new tool on bin packing // Ann. Oper. Res.— 1988.— 16, № 1.— P. 327—332.
31. Шеннон Р. Ю. Имитационное моделирование систем — искусство и наука / Пер. с англ.— М.: Мир, 1978.
32. Имитационное моделирование в оперативном управлении производством.— М.: Машиностроение, 1984.— 208 с.
33. Нейлор Т., Ботон Дж., Бердик Д. и др. Машинные имитационные эксперименты с моделями экономических систем / Пер. с англ.— М.: Мир, 1975.— 504 с.
34. Arthur J. L. et al. Microcomputer simulations systems // Comput. and Oper. Res.— 1986.— 13, № 2—3.— P. 167—183.
35. Гончарова Л. Н., Корягин Д. А. Язык заданий пакета для построения имитационных систем / Препр. № 180. ИПМ АН СССР, 1986.
36. Родионов А. С. Интеллектуальное моделирование — новое направление в системах имитации. Обзор последних публикаций // Экспертные системы и анализ данных.— Новосибирск: Наука, 1988.— С. 19—35.
37. Schengilli J. Optimal scheduling // Proc. 2nd Int. Conf. Simul. Manuf. Chicago, Ill. 24—26 June 1986.— Kempston, 1986.— P. 139—145.
38. Grant H. Product scheduling using simulation technology // Ibid.— P. 129—138.
39. Blackwell R. Q. A discrete event scheduler in a dynamic production system // Proc. of the 1986 Winter Simulation Conf. Washington, D. C. 8—10 Dec. 1986.— New York, N. Y. 1986.— P. 661—664.
40. Bell P. C. Visual interactive modelling in operational research: successes and opportunities // J. Oper. Res. Sci.— 1985.— 36, № 11.— P. 975—982.
41. Ньюэлл А., Саймон Г. GPS-программа, моделирующая процесс человеческого мышления // Вычислительные машины и мышление.— М.: Мир, 1967.— С. 283—301.
42. Encycl. of Artif. Intell. / Ed. Shapiro S. C.— N. Y.: Wiley, 1987.
43. Kautz H. A., Pednault E. P. D. Planning and plan recognition. // AT and T Tech. J.— 1988.— 67, № 1.— P. 25—40.
44. Tate A. A review of knowledge-based planning techniques // Expert Syst.'85. Proc. 5th Techn. Conf. Brit. Comput. Soc. Spec. Group Expert Syst. Warwick 17—19 Dec. 1985.— Cambridge e. a., 1985.— P. 89—111.
45. Нильсон Н. Искусственный интеллект. Методы поиска решений / Пер. с англ.— М.: Мир, 1973.— 270 с.
46. Fikes R. E., Nilsson N. J. STRIPS: a new approach to application of theorem proving to problem solving // Artif. Intell.— 1971. 2, № 2.— P. 189—208.
47. Korj R. E. Planning as search: a quantitative approach // Ibid.— 1987.— 33, № 1.— P. 65—88.
48. Sacerdoti E. D. Planning in a hierarchy of abstract spaces // Ibid.— 1974.— 5, № 2.— P. 115—135.
49. Stefic M. Planning with constraints (MOLGEN: P.1) // Ibid.— 1981.— 16, № 2.— P. 111—140.
50. Stefic M. Planning and metapanning (MOLGEN: P. 2) // Ibid.— P. 141—169.
51. Tate A. Generating project networks // Proc. 5th Int. Joint Conf. on Artif. Intell.— Boston: MIT, 1977.— P. 237—253.
52. Нильсон Н. Принципы искусственного интеллекта / Пер. с англ.— М.: Радио и связь, 1985.— 376 с.
53. Sacerdoti E. D. A structure for plans and behavior.— N. Y.: Elsevier, 1977.
54. Fox B. R. The implication of opportunistic scheduling // Intell. Autom. Syst.: Int. Conf., Amsterdam, 8—11 Dec. 1986.— Amsterdam e. a., 1987.— P. 231—240.
55. Wilkins D. E. Representation in a domain independent planner // Proc. Int. Joint Conf. on Artif. Intell.— Karlsruhe, 1983.— P. 733—740.
56. Vere S. Planning in time: windows and durations for activities and goals // IEEE Trans. on pattern analysis and machine intelligence.— 1983.— PAMI-5, № 3.— P. 246—267.
57. Fox M. S., Smith S. F. ISIS: a knowledge based system for factory scheduling // Expert. Syst.— 1984.— 1, № 1.— P. 25—49.
58. Попов Э. В. Экспертные системы: Решение неформализованных задач в диалоге с ЭВМ.— М.: Наука, 1987.— 288 с.
59. Хейес-Рот Ф., Уотерман Д., Ленат Д. Построение экспертных систем.— М.: Мир, 1987.— 430 с.
60. Уотерман Д. Руководство по экспертным системам / Пер. с англ.— М.: Мир.— 1989.— 388 с.
61. Пелавин Р., Аллен Дж. Ф. Формальная логика планов в задачах, богатых временными связями // ТИИЭР.— 1986.— 74, № 10.— С. 79—100

62. *Allen J. F.* Maintaining knowledge about temporal intervals // *Comm. ACM.*— 1983.— 26, № 11.— P. 832—843.
63. *McDermott D. V.* A temporal logic for reasoning about processes and plans // *Cogn. Sci.*— 1982.— 6.— P. 101—155.
64. *Allen J. F.* Towards a general theory of action and time // *Artif. Intell.*— 1984.— 23, № 2.— P. 123—154.
65. *Минский М.* Фреймы для представления знаний / Пер. с англ.— М.: Энергия.— 1979.
66. *Fox M. S.* Knowledge representation for decision support. // *Knowledge representation for decision support systems* / Ed. Methlie L. B., Sprague R. H.— N. Y.: Elsevier, 1985.— P. 3—26.
67. *Mackerle J.* A review of expert systems development tools // *Eng. Comput.*— 1989.— 6, № 1.— P. 2—17.
68. *Ларичев О. И., Петровский А. Б.* Системы поддержки принятия решений. Современное состояние и перспективы // *Итоги науки и техники. Сер. Техн. киберн.*— М.: ВИНТИ.— 1987.— Т. 21.— С. 131—164.
69. *Lerape C.* SOJA: a daily workshop scheduling system // *Expert System'85 Proc. 5th Techn. Conf. Brit. Comput. Soc. Spec. Group Expert Syst Warwick*, 17—19 Dec. 1985.— Cambridge, e. a., 1985.— P. 195—211.
70. *Kanai N., Yokoi S.* et al. An expert system to assist production planning // *Proc. Int. Workshop on Artif. Intell. for Ind. Appl. Hitachi City*, 25—27 May 1988.— New York (N. Y.), 1988.— P. 220—224.
71. *Brown M. C.* The dynamic rescheduler: conquering the changing production environment // *4th Conf. Artif. Intell. Appl., San-Diego, Calif.* 14—18 March 1988.— Proc. Washington (D. C.), 1988.— P. 175—180.
72. *Shen S., Chang Y. L.* Schedule generation in a flexible manufacturing systems: a knowledge-based approach // *Decision Support Systems.*— 1988.— 4.— P. 157—166.
73. *Ammons J. C., Govindaraj T., Mitchell C. M.* Decision models for aiding FMS scheduling and control // *IEEE Trans. Syst. Man and Cybern.*— 1988.— 18, № 5.— P. 744—756.
74. *Ben-Arieh D., Moodie C. L.* Knowledge based routing and sequencing for discrete part production // *J. Manuf. Syst.*— 1988.— 6, № 4.— P. 287—297.
75. *Suranjan De., Lee A.* Making scheduling decisions in an FMS using the state-operator framework in AI // *Proc. IEEE Int. Conf. Syst., Man and Cybern.* 14—17 Oct. 1986 v. 2.— Atlanta (Ga), 1986.— P. 1178—1181
76. *Wu S Y. D., Wysk R. A.* Multi-pass expert control system—a control / scheduling structure for flexible manufacturing cells // *J. Manuf. Syst.*— 1988.— 7, № 2.— P. 107—120.
77. *Bruno G., Elia A., Laface P.* A rule-based system to schedule production // *Computer.*— 1986.— 19, № 7.— P. 32—39.
78. *Newman P. A., Kempf K. G.* Opportunistic scheduling for robotic machine tending // *Artif. Intell. Appl. Eng. Knowledge-based syst. Proc. 2nd Conf. Miami Beach, (Fla.)* 11—13 Dec. 1985.— Amsterdam, 1985.— P. 168—173.
79. *Chen D.-S., Kallsen A., Shider R. C.* School bus routing and scheduling: an expert system approach // *Comput. Ind. Eng.*— 1988.— 15.— P. 179—183.
80. *Husain N., Reddy Y. V.* Continuous mileage vehicle scheduling as rule-based search // *Artif. Intell. Expert Syst. and Languages Modell. and Simul. Proc. 1st IMACS Symp., Barselona*, 2—4 June 1987.— Amsterdam etc., 1988.— P. 129—135.
81. *Tsuruta S., Matsumoto K. A.* Knowledge-based interactive train scheduling system aiming at large-scale complex planning expert systems // *Proc. Int. Workshop on Artif. Intell. for Ind. Appl. Hitachi City*, 25—27 May 1988.— New York (N. Y.) 1988.— P. 490—495.
82. *Brazile R. P., Swigger K. M.* GATES—an airline gate assignment and tracking expert system // *IEEE Expert.*— 1988.— 3, № 2.— P. 33—39.
83. *Schweizle J. G., Zoch D. R.* A hierarchical planning system for mission support // *Expert. Syst. Gov. Symp. McLean (Va)*, 22—24 Oct. 1986.— Washington (D. C.), 1986.— P. 220—224.
84. *Hankins G. B., Jordan J. W.* et al. EMPRESS—expert mission planning and replanning scheduling system // *Ibid.*— P. 170—175.
85. *Podbury C. A., Dillon T. S.* Maintenance scheduler: a frame-based expert system // *7th Int. Workshop Expert Syst. and Appl., Avignon* 13—15 May 1987.— Paris, 1987.— 1.— P. 650—666.
86. *Steffen M. S., Green T. J.* An application of hierarchical planning and constraint-directed search to scheduling parallel processors // *Ibid.*— 2.— P. 910—917.
87. *Bensana E., Correge M. et al.* An expert system approach to industrial job-shop scheduling // *Proc. IEEE Conf. Rob and Autom., San Francisco (Calif.)*, 7—10 Apr. 1986.— Washington (D. C.), 1986.— 3.— P. 1645—1650.
88. *Erschter J., Esquirol P.* Decision-aid in job-shop scheduling: a knowledge-based approach // *Ibid.*— P. 1651—1656.
89. *Lamatsch A., Morlock M et al.* SCHEDULE-an expert-like system for machine scheduling // *Ann. Oper. Res.*— 1988.— 16.— P. 425—438.
90. *Monfroglio A.* Timetabling through a deductive database // *Data and Knowledge Engng.*— 1988.— 3, № 1.— P. 1—27.
91. *O'Keefe R.* Simulation and expert systems—a taxonomy and some examples // *Simulation.*— 1986.— 46, № 1.— P. 10—15.
92. *Schannon R. E., Mayer R., Adelsberger H. H.* Expert systems and simulation // *Ibid.*— 1985.— 44, № 6.— P. 275—284.
93. *Кларк К., Маккейб Ф.* Введение в логическое программирование / Пер. с англ.— М.: Радио и связь, 1987.— 227 с.
94. *Bullerson W. T., Schultz C. R.* Production rule-based Simulation for job-shop scheduling // *Proc. Summer Comput. Simuel. Conf. Reno (Nev.)* 28—30 July, 1986.— San. Diego, (Calif.), 1986.— P. 718—723.
95. *Hayes-Roth B.* A blackboard architecture for control // *Artif. Intell.*— 1985.— 26, № 3.— P. 251—321.
96. *Корбут А. А., Сигал И. X., Финкельштейн Ю. Ю.* Гибридные методы в дискретной оптимизации // *Изв. АН СССР. Сер. Техн. Кибернетика.*— 1988.— № 1.— С. 65—77.
97. *Schroer B. J., Tseng F T.* An intelligent assistant for manufacturing system simulation // *Int. J. Prod. Res.*— 1989.— 27, № 10.— P. 1665—1683.

УДК 681.51

НЕЧЕТКИЕ МОДЕЛИ УПРАВЛЕНИЯ ДИНАМИЧЕСКИМИ СИСТЕМАМИ

Р. А. Алиев, Э. Г. Захарова, С. В. Ульянов

ВВЕДЕНИЕ

Обзор посвящен вопросам анализа и синтеза нечетких моделей управления сложными динамическими системами и является продолжением работ [1—5]. При этом основное внимание уделяется прикладным аспектам теории нечетких моделей, алгоритмов управления и САУ. Основные положения теорий нечетких множеств и нечетких алгоритмов управления и принятия решений читатель может найти в [6—19] и др.

Описание и анализ работоспособности внедренных алгоритмов и систем управления на базе нечеткой логики (лингвистических переменных) позволяют глубже оценить целесообразность, предельные возможности используемых подходов и содержательную интерпретацию результатов теории нечетких

моделей. Перечень конкретных примеров нечетких алгоритмов и САУ приведен в разделе 2.

Изложение материала базируется на введенных Л. А. Заде принципах нечеткой логики (см. раздел 1). При этом предполагается, что совокупность управляющих алгоритмов и САУ принадлежит к классу нечетких систем. Состояние сложной системы и управляющие воздействия рассматриваются как лингвистические переменные, а выбор конкретных значений управления осуществляется на основе композиционного правила вывода. В результате неточность (нечеткость) описания динамического поведения объекта управления, возникающая из-за лингвистической аппроксимации описания передаточных функций системы, компенсируется более высоким по уровню алгоритмом управления благодаря учету (помимо количественных) качественных признаков динамического поведения объекта управления на основе лингвистических переменных.

Приводятся оценки сложности внедренных алгоритмов управления и сравнительные оценки с другими известными алгоритмами управления.

Предварительно рассмотрим пример применения теории нечетких моделей к традиционным задачам синтеза структур регуляторов. Дополнительные вопросы теории нечетких моделей динамических систем рассмотрены в [20—26] и др.

Пример. Допустим, что заданы пространство входов (U, ρ) и выходов (Y, d) с соответствующими элементами $u \in U$, $y \in Y$ и метриками $\rho(x_1, x_2)$ и $d(y_1, y_2)$. Предполагается, что модель «вход—выход» управляемой системы описывается отображением $y = f(u)$ из U в Y . Реальная модель управляемой системы содержит множество неопределенностей в определении параметров структуры, неточностей измерений выходного процесса y и т. п. Предлагается, согласно [27], учитывать подобного рода НЕ-факторы [28] путем введения нечеткого отношения $\mu_\varepsilon(y, z)$ для выходных сигналов $y, z \in Y$, которое может интерпретироваться как степень принадлежности пары сигналов (y, z) отношению $d(y, z) < \varepsilon$ для неотрицательного действительного числа ε . Таким образом, неравенство $d(y, z) < \varepsilon$ характеризует влияние неучтенных факторов в модели на отклонение выходных сигналов с заданной функцией принадлежности $\mu_\varepsilon(y, z)$.

На функцию принадлежности $\mu_\varepsilon(y, z)$ необходимо наложить естественные условия, которые отражали бы физические требования к динамике управляемой системы. Поскольку функция μ_ε есть отношение, то она рассматривается как отображение $Y \times Y$ в интервал $[0, 1]$; если $\varepsilon = 0$, т. е. требуется совпадение сигналов y и z , а сами сигналы произвольны в Y , то естественно принять $\mu_0(y, z) = 0$; аналогично, $\mu_\varepsilon(y, z) = \mu_\varepsilon(z, y)$; если известно, что $y = z$ для всех ε , то $\mu_\varepsilon(y, z) = 1$; если $\varepsilon \leq \eta$, т. е. требования на точность выше для ε , чем для η ,

то $\mu_\varepsilon(y, z) \leq \mu_\eta(y, z)$; если асимптотически увеличивать $\varepsilon \rightarrow \infty$, т. е. снижать требования на точность, то $\lim_{\varepsilon \rightarrow \infty} \mu_\varepsilon(y, z) = 1$. Ана-

логично, для двух критериев точности ε и η с соответствующими $\mu_\varepsilon(x, y)$ и $\mu_\eta(y, z)$ имеет место естественное условие $\mu_{\varepsilon+\eta}(x, z) \geq \min(\mu_\varepsilon(x, y), \mu_\eta(y, z))$ для любых сигналов x, y, z из пространства выходов Y . Таким образом, приведенное описание модели оперирует пространством (Y, μ) , но уже не в терминах расстояния $d(y, z)$, а в терминах нечеткого отношения $\mu_\varepsilon(y, z)$, включающего в себя неучтенные факторы неопределенности в модели $y = f(u)$, которые влияют на отклонение выходного сигнала y . Тогда топология пространства (Y, μ) может быть введена с помощью понятия t -окрестности выхода системы в виде $N_y(t) = \{z : \mu_t(y, z) > 1-t\}$, а окрестность выходных сигналов в пространстве Y — как объединение $\bigcup_{y \in Y} \{N_y(t) \mid t > 0\}$.

Задачу управления можно сформулировать следующим образом: по заданным t и y_0 найти управление U , такое, чтобы y , определяемое из уравнения $y = f(u)$, находилось в t -окрестности $N_{y_0}(t)$. Существование такого управления может быть доказано из определения понятия t -управляемости системы в точке y_0 : система $y = f(u)$ является t -управляемой в точке $y_0 \in Y$, если существует такое управление $u \in U$, для которого $f(u) \in N_{y_0}(t)$.

В терминах функции принадлежности можно наложить ограничения на динамику управляемой системы в виде аналогов условий Липшица: существуют положительная константа C и убывающая функция $V: \mathcal{R} \rightarrow [0, 1]$ (со свойствами $V(0) = 0$ и $\lim_{\varepsilon \rightarrow \infty} V(\xi) = 1$), такие, что $\mu_{\varepsilon+\rho}(f(u), f(v)) \geq V(\varepsilon - \rho(u, v))$. Будем называть данное определение условиями (C, V) -Липшица. Напомним, что $\rho(u, v)$ — расстояние в пространстве входов, т. е. управлений U и $(u, v) \in U$.

По аналогии, отображение g из Y в Y будет оператором сжатия в пространстве (Y, μ) , если существует действительное число $k \in (0, 1)$, такое, что выполняется условие $\mu_\varepsilon(g(y), g(z)) \geq \mu_{\varepsilon/k}(y, z)$. Согласно [27], если g — оператор сжатия в компактном пространстве (Y, μ) , то отображение g имеет неподвижную точку, т. е. существует единственное $y^* \in Y$, для которого $g(y^*) = y^*$. Более того, для любого $z \in Y$ последовательность итераций для отображения g типа $z_{n+1} = g(z_n)$ будет сходиться к неподвижной точке.

Рассмотрим теперь применение принципа неподвижной точки при проектировании нечеткого регулятора, описываемого отображением r из пространства (Y, μ) в пространство входов (U, ρ) , т. е. $u = r(y)$, генерирующее детерминированный сигнал управления u из нечеткого выходного сигнала y . Согласно [27], нечеткая система, для которой выполняются аналог условий (C, V) -Липшица, t -управляемости в точке y_0 , а выходные сигнала

лы $y \in (Y, \mu)$, является t -устойчивой в точке y_0 , т. е. существует нечеткий контроллер $r(y) = u$, который переводит выходной сигнал (состояние) y в t -окрестность точки y_0 .

Действительно, условие t -управляемости обеспечивает существование такого управления u_0 , что $f(u_0) = y_0$, $y_0 \in N_{y_0}(t)$. Предположим, что нечеткий контроллер выбран именно таким образом, что для $y = y_0$ имеется u_0 , т. е. $r(y_0) = u_0 = f^{-1}(y_0)$. Тогда y_0 есть неподвижная точка отображения $f(r(\cdot))$. Действительно, имеем $f(r(y_0)) = y_0 \in N_{y_0}(t)$. Неподвижную точку y_0 можно рассматривать как точку притяжения оператора сжатия в пространстве (Y, μ) . Тогда можно получить необходимые условия для нечеткого контроллера следующего вида. Допустим, что существует такая константа $\rho > 0$ и $cp < 1$. Для произвольных $(y, z) \in Y$ и неотрицательной действительной ϵ контроллер $r(y)$ должен удовлетворять условию $V(\rho\epsilon - \rho(r(y), r(z))) \geq \mu_\epsilon(y, z)$. Учитывая одновременно свойство (c, V) -Липшица системы $y = f(u)$, получим, что $\mu_{c\epsilon}(f(r(y)), f(r(z))) \geq \mu_{c\epsilon/k}(y, z)$, где $k = cp < 1$. Таким образом, для каждого выходного сигнала $(y, z) \in Y$ и неотрицательной величины ϵ получили условия для оператора сжатия, а используя результат о существовании для него неподвижной точки y_0 , — и доказательство единственности точки притяжения в виде неподвижной точки. Учитывая, что сама точка y_0 лежит в своей t -окрестности, получаем доказательство существования такого нечеткого контроллера $r(y_0) = u_0$.

По аналогии с вариантом существования неподвижной точки для операторов сжатия, зависящих от параметра, можно рассмотреть управляемую систему $y = f(u, w)$, в которую введены нечеткие параметры w (например, нечеткие числа $(L-R)$ -типа или в параметрическом виде) в параметрическом пространстве (W, ν) . Отображение f переводит декартово произведение $U \times W$ метрического пространства входных сигналов (U, ρ) и параметрического пространства (W, ν) в пространство выходных сигналов (Y, μ) . Нечеткое отношение ω для декартового произведения $U \times W$ может быть определено как $\omega_\epsilon((u_1, w_1), (u_2, w_2)) = \min(1(\epsilon - \rho(u_1, u_2)), \nu_\epsilon(w_1, w_2))$, где 1 — единичная функция. При этом предполагается, что существует обратное отображение вида $w = f^{-1}(\cdot, y)$, что означает ν -идентификацию системы. Если редуцировать параметрическое пространство к метрическому пространству (W, d_w) , то системе можно идентифицировать полностью в смысле определенных метрических пространств. Такой подход особенно успешно используется при проектировании нечетких адаптивных контроллеров для нечетких систем управления, когда управление и измерение текущих параметров используются в параллельном режиме.

Тогда, используя понятие t -окрестности и принцип неподвижной точки, можно осуществить процесс проектирования нечетких регуляторов и для таких вариантов нечетких систем с пространством нечетких параметров в структуре.

В данном разделе рассмотрены основные идеи нечеткой логики, которая лежит в основе процессов разработки моделей объектов управления на основе лингвистической аппроксимации ее качественного описания динамического поведения и разработки соответствующих нечетких алгоритмов (инструкций) управления для нечетких контроллеров.

Основная цель изложенного материала заключается в ознакомлении читателя с некоторыми прикладными аспектами методов нечеткой логики и иллюстрации основных идей на конкретных примерах. Более подробное и строгое изложение затрагиваемых вопросов и проблем можно найти в [2, 6—19, 29—33].

Основные положения булевой логики [34—36]. Согласно одному из самых распространенных определений [36], логика есть анализ точных методов рассуждения. При этом анализируется, в первую очередь, форма, а не содержание доводов в том или ином рассуждении. Истинность или ложность отдельных посылок или заключений не интересует исследователя: интересует лишь, вытекает ли истинность заключения из истинности посылок. Систематическая формализация и составление каталогов правильных способов рассуждений — одна из основных задач логики.

Из простых высказываний путем соединения их различными способами в логике можно составить новые, более сложные высказывания. В дальнейшем будем рассматривать истинностно-функциональные комбинации, в которых истинность или ложность новых высказываний определяется истинностью или ложностью составляющих высказываний.

Одной из простейших операций над высказываниями является отрицание. Например, если A есть высказывание, то отрицание A обозначается, как известно, $\neg A$ и читается «не A ».

Другой истинностно-функциональной операцией над высказываниями является конъюнкция «и», которая обозначается в виде $A \& B$, т. е. A и B . Высказывание истинно тогда и только тогда, когда истинны оба высказывания A и B . Высказывания A и B называются конъюнктивными членами конъюнкции $A \& B$.

Операция дизъюнкции над высказываниями A и B соответствует связке «или» и обозначается $A \vee B$. Следует отметить, что в обычном языке связка «или» употребляется в двух различных смыслах: разделительном и соединительном. В операции дизъюнкции связка «или» имеет соединительный смысл.

Следующей важной истинностно-функциональной операцией является следование: «если A то B ». Это высказывание ложно, когда посылка A истинна, а заключение B ложно. Обозна-

чение высказывания «если A то B » следующее: $A \supset B$ Это выражение называется импликацией.

Выражение « A тогда и только тогда, когда B » обычно обозначается через « $A \equiv B$ ». Такое выражение называется эквивалентностью. Очевидно, $A \equiv B$ истинно тогда и только тогда, когда A и B имеют одно и то же истинностное значение.

Ниже в качестве примера описана истинностная таблица (табл. 1.1) для всех приведенных операций над высказываниями (здесь $И$ — истинно, $Л$ — ложно).

Таблица 1.1

A	B	A	$A \& B$	$A \vee B$	$A \supset B$	$A \equiv B$
И	И	Л	И	И	И	И
Л	И	И	Л	И	И	Л
И	Л	Л	Л	И	Л	Л
Л	Л	И	Л	Л	И	И

Символы \neg , $\&$, \vee , \supset , \equiv называются пропозициональными связками. Всякое высказывание, построенное при помощи этих связок, имеет некоторое истинностное значение, зависящее от истинности значений составляющих высказываний.

Пропозициональной формой называется выражение построенное из пропозициональных букв A , B , C и т. д. с помощью пропозициональных связок.

Всякая пропозициональная форма определяет некоторую истинностную функцию, которая графически может быть представлена истинностной таблицей для этой пропозициональной формы. Истинностной функцией от n аргументов называется всякая функция от n аргументов, принимающая истинностные значения $И$ и $Л$, если аргументы ее пробегают те же значения.

Пропозициональная форма, которая истинна независимо от того, какие значения принимают встречающиеся в ней пропозициональные буквы, называется тавтологией. Пропозициональная форма является тавтологией тогда и только тогда, когда соответствующая истинностная функция принимает только значение $И$.

Например, следующие предложения являются пропозициональными тавтологиями: 1) $\neg(A \& \neg A)$ — закон отрицания противоречия; 2) $((A \vee B) \rightarrow (\neg A \rightarrow B))$ — выражение дизъюнкции через отрицание и импликацию; 3) $((A \rightarrow B) \& A) \rightarrow B$.

В свою очередь, пропозициональная форма, которая ложна при всех возможных истинностных значениях ее пропозициональных букв, называется противоречием. Примером противоречия может служить пропозициональная форма $(A \equiv \neg A)$ или $(A \& (\neg A))$.

Импликация, как отмечалось, имеет следующее важное свойство, называемое правилом отделения (*modus ponens*): если $(A \rightarrow B)$ истинно и A истинно, то B истинно. Иначе это правило называется первой формой гипотетического силлогизма. Под силлогизмом подразумевается дедуктивное умозаключение, в котором одно суждение является необходимым следствием двух других. Это свойство, как уже указывалось, играет важную роль при моделировании сложных технологических процессов (см. раздел 2).

Согласно [37, 38], логика — исчисление, синтаксические конструкции которого разбиты на классы, в каждом классе выделен подкласс элементарных понятий и определена операция подстановки понятий B_1, \dots, B_n вместо элементарных понятий E_1, \dots, E_n тех же классов в конструкции A , обозначаемая $A[E_1, \dots, E_n | B_1, \dots, B_n]$. Если подстановка дает синтаксически корректную конструкцию, то она принадлежит тому же классу, что и A . Исчисление должно обладать следующим свойством подстановки. Пусть Π — вывод B из A_1, \dots, A_k и $A_i[E_1, \dots, E_n | C_1, \dots, C_n], \dots, A_k[E_1, \dots, E_n | C_1, \dots, C_n], B_i[E_1, \dots, E_n | C_1, \dots, C_n]$ — корректные подстановки. Тогда, осуществляя те же подстановки в выводе Π , получим вывод $B[E_1, \dots, E_n | C_1, \dots, C_n]$ и $A_1[\cdot | \cdot], \dots, A_k[\cdot | \cdot]$.

Классическое исчисление предикатов удовлетворяет этому свойству, поскольку оно содержит правило подстановки. Таким образом, в логике не содержится конкретных понятий, элементарное понятие рассматривается как свободная переменная, вместо которой можно произвести подстановку. При этом последовательность логического вывода сохраняется. Логический язык традиционного, истиннозначного типа, в котором описываются свойства объектов и планов построения, часто называется дескрипторным.

Логику вместе с ее классом моделей и отображением, в которой каждой ее формуле сопоставляется истинностное значение в модели таким образом, что эквивалентные формулы получают одинаковые истинностные значения называют дескриптивной логикой [32, 36]. Понятие дескриптивной логики является относительным, так как логика считается в данном контексте дескриптивной, если она используется лишь для описания свойств предметов и их преобразований. Следовательно, выводимость в дескриптивной логике является лишь способом оценки значения выведенной формулы, а способом ее вывода в принципе не интересуются.

Логика, в которой каждому рассуждению соответствует мысленное построение, часто называется конструктивной логикой. В классической логике препятствиями для подобного рода построений являются закон исключения третьего $A \vee \neg A$ и закон двойного отрицания $\neg \neg A \Rightarrow A$ С точки зрения конструктивной логики закон исключения третьего называется принци-

пом всезнания, а закон двойного отрицания — принципом универсального решателя задач [13, 36—39].

Рассмотрим некоторые дополнительные модели конструктивной логики, применяя принцип изменения аксиом.

Нечеткая логика. Эта логика (называемая также многозначной) основывается на теории нечетких множеств, а пространством истинности в этой логике является действительный интервал $[0,1]$. Определим семантическую функцию истинности многозначной логики. Пусть P является высказыванием, а $v(p)$ — его значением истинности, при этом $v(P) \in [0, 1]$.

Значение отрицания для высказывания P определяется в виде $v(\neg P) = 1 - v(P)$. Следовательно, $v(\neg \neg P) = v(P)$.

Связка импликация \rightarrow определяется следующим образом: $v(P \rightarrow Q) = v(\neg P \vee Q)$, а эквивалентность как

$$v(P \leftrightarrow Q) = v[(P \rightarrow Q) \wedge (Q \rightarrow P)].$$

Тавтология и противоречие соответственно определяются следующим образом: $v(\dot{P}) = v(P \vee \neg P)$; $v(\ddot{P}) = v(P \wedge \neg P)$. Более обобщенно $v(\dot{P}Q) = v((P \vee \neg P) \vee (Q \vee \neg Q))$; $v(\ddot{P}Q) = v((P \wedge \neg P) \wedge (Q \wedge \neg Q))$.

Поскольку в зависимости от способов введения операций объединения и пересечения существует как минимум 3 варианта теории нечетких множеств [2, 19], то соответственно можно получить 3 варианта описания нечетких логик. В табл. 1.2 приведены, согласно [2, 5, 19], три таких способа описания нечетких логик. Следовательно, нечеткая логика с максиминными операциями ($\mathcal{F}(X)$, \cup , \cap , $-$), ограниченными операциями ($\mathcal{F}(X)$, \wedge , \vee , $-$) и ($\mathcal{F}(X)$, $\dot{+}$, \cdot , $-$), которую часто называют вероятностной логикой [5, 19, 32], может быть описана в терминах соответствующих операций, а ее алгебраические свойства определяются свойствами применяемых связок.

Многозначная логика, согласно [17], является размытием (в смысле нечеткости) стандартного исчисления высказываний (в смысле принципа обобщения). В этой логике каждому высказыванию P ставится в соответствие нормализованное нечеткое множество в $[0, 1]$, т. е. пара $[\mu_P(0), \mu_P(1)]$ интерпретируется как степень ложности и степень истинности соответственно. Поскольку логические связки стандартного исчисления высказываний являются функционалами истинности, т. е. представляются в виде функций, то их можно «размыть». Следует отметить, что логика подобного типа была рассмотрена независимо Клином и Дайнесом.

Если истинность $v(P)$ высказывания P трактовать как лингвистическую переменную [2, 5, 19], то придем к нечеткой логике с лингвистическими значениями типа истинный, очень истинный, более или менее истинный и т. д. Такую нечеткую логику называют «нечеткозначной ло-

Таблица 1.2

Название связки	Обозначение связки	Нечеткая логика с максиминными операциями	Нечеткая логика с ограниченными операциями	Вероятностная нечеткая логика
Тавтология	\dot{P}	$\max(p, 1-p)$	1	$1-p(1-p)$
Противоречие	\ddot{P}	$\min(p, 1-p)$	0	$p(1-p)$
Отрицание	$\neg P$	$1-p$	$1-p$	$1-p$
Дизъюнкция	$P \vee Q$	$\max(p, q)$	$\min(1, p+q)$	$p+q-p \cdot q$
Конъюнкция	$P \wedge Q$	$\min(p, q)$	$\max(0, p+q-1)$	$p \cdot q$
Импликация	$P \rightarrow Q$	$\max(1-p, q)$	$\min(1, 1-p+q)$	$1-p+p \cdot q$
Эквивалентность	$P \leftrightarrow Q$	$\min[\max(1-p, q), \max(p, 1-q)]$	$1- p-q $	$(1-p+p \cdot q)(1-q+q \cdot p)$
Штрих Шеффера	$P Q$	$\max(1-p, 1-q)$	$\min(1, 1-p+1-q)$	$1-pq$
Исключающее ИЛИ	$P \oplus Q$	$\max[\min((1-p), q), \min(p, 1-q)]$	$ p-q $	$1-(1-p+pq)(1-q+q \cdot p)$
Стрелка Пирса	$P \downarrow Q$	$\min(1-p, 1-q)$	$\max(0, 1-p-q)$	$(1-p)(1-q)$

гикой» и она составляет основу для теории приближенных суждений [2, 9, 33]. Нечеткозначная логика описывается нечеткими множествами типа 2, у которых функции принадлежности являются нечеткими числами [2, 5, 19].

Семантические правила вычисления функций истинности для отрицания, конъюнкции и дизъюнкции запишутся в следующем виде: $v(\neg P) = 1 \ominus v(P) = \text{ant } v(P)$; $v(P \wedge Q) = \min(v(P), v(Q))$; $v(P \vee Q) = \max(v(P), v(Q))$. Тогда можно получить в терминах нечетких множеств типа 2 семантические правила для логических связей нечетких логик из табл. 1.2. Некоторые из таких семантических правил приведены в табл. 1.3.

Рассмотрим в качестве иллюстрации логические связи из табл. 1.3 применительно к некоторым лингвистическим переменным истинности.

Пример [19]. Допустим, что P — сомнительно, Q — истинно. Рассмотрим применение нечеткой логики с максимными операциями. Для импликации из позиция 1 табл. 1.3 следует $v(P \rightarrow Q) = \max(\text{ant (сомнительно)}, \text{истинно}) \approx \text{истинно}$; для импликации $v(Q \rightarrow P) = \max(\text{ant (истинно)}, \text{сомнительно}) \approx \text{сомнительно}$; для операции эквивалентности из позиции 2 табл. 1.3 имеем $v(P \leftrightarrow Q)$ — сомнительно.

Аналогично, для нечеткой логики с ограниченными операциями имеем: $v(P \rightarrow Q) = \min(1, \text{ant (сомнительно)} \oplus \text{истинно}) = \min(1, \text{сомнительно} \oplus \text{истинно})$. Таким образом, если операция даст значение истинности ≥ 1 , то $v(P \rightarrow Q) = 1$. В общем случае, $v(P \rightarrow Q)$ с истинно и совпадает с обычной импликацией. Полагаем, что приведенного примера достаточно для понимания принципиальных отличий описанных вариантов нечеткой логики.

Следует отметить, что теории нечетких логик из табл. 1.3 не являются единственными из известных нечетких логик. Развитие теории приближенных суждений и нечеткого вывода, а также теории нечетких алгоритмов привело к необходимости анализа и синтеза, в первую очередь, моделей нечетких логик, связанных с импликацией [2, 19, 29, 39—42]. Это объясняется также тем обстоятельством, что в последние годы получил широкое развитие подход к представлению знаний об элементах систем и объектов управления в виде совокупности правил вывода [43, 13, 44] с использованием имитационного моделирования динамического поведения систем. Оказалось [45], что в результате логической имитации автоматически формируется база знаний в виде системы продукций «ЕСЛИ [(начальная ситуация) и (стратегия)], ТО [(характер глобальной регулярности)]», т. е. в виде операции импликации. При этом алго-

ритм имитационного моделирования корректен в смысле нечеткого вывода [45] и семантической эквивалентности [46].

В связи с этим представляется целесообразным дать семантический анализ основных многозначных (нечетких) логик и выделить наиболее гибкие из них в смысле прикладного применения, в частности для использования в нечетких алгоритмах и нечетких регуляторах (см. раздел 2).

Аналогично тому, как строились нечеткие логики из табл. 1.2 в соответствии с моделями теории нечетких множеств, анализ многозначных логик следует начинать с изучения свойств вариантов теории нечетких множеств — теории нечетких «сильных» множеств, полученной изменением (усилением) операции импликации и др. Применяя принцип изменения аксиом [47—49], можно получить различные варианты теории нечетких «сильных» множеств [19, 29, 40, 41]. При этом могут быть самые различные критерии и меры близости между различными типами импликации [42].

Допустим, что A и B являются нечеткими подмножествами четкого универсального множества U . В теории нечетких множеств считается [2, 19], что A является подмножеством B тогда и только тогда (iff), когда $\mu_A \leq \mu_B$, т. е. $\forall x \in U$ имеем $\mu_A(x) \leq \mu_B(x)$.

Таблица 1.3

№ п/п	Вид логической связи	Вид нечеткой логики	
		Нечеткая логика с максимными операциями, $\{\mathcal{F}(x), \cup, \cap, \neg\}$	Нечеткая логика с ограниченными операциями, $\{\mathcal{F}(x), \vee, \wedge, \neg\}$
1	Импликация	$v(P \rightarrow Q) = \max(1 \ominus v(P), v(Q))$	$v(P \rightarrow Q) = \min(1, 1 \ominus v(P) \oplus v(Q))$
2	Эквивалентность	$v(P \leftrightarrow Q) = \min[\max(1 \ominus v(P), v(Q)), \max(1 \ominus v(Q), v(P))]$	$v(P \leftrightarrow Q) = 1 \ominus v(P) \ominus v(Q) $
3	Исключающее ИЛИ	$v(P \text{ ex } Q) = \max[\min(1 \ominus v(P), v(Q)), \min(1 \ominus v(Q), v(P))]$	$v(P \text{ ex } Q) = v(P) \ominus v(Q) $
4	Тавтология	$v(\dot{P}) = \max(v(P), 1 \ominus v(Q))$	$v(\dot{P}) = 1$
5	Противоречие	$v(\dot{P}) = \min(v(P), 1 \ominus v(Q))$	$v(\dot{P}) = 0$

№ ст/п	Наименование много- значной логики	Обозначение импликации	Интерпретация операции «импликация»
1	S*-логика	$a \rightarrow b$ S^*	$a \rightarrow b = \begin{cases} 1, & \text{если } a \neq 1, \text{ или } b = 1 \\ 0, & \text{в противном случае} \end{cases}$
2	S-логика («стан- дартная после- довательность»)	$a \rightarrow b$ S	$a \rightarrow b = \begin{cases} 1, & \text{если } a \leq b \\ 0, & \text{в противном случае} \end{cases}$
3	G-логика («Гедель- манская после- довательность»)	$a \rightarrow b$ G	$a \rightarrow b = \begin{cases} 1, & \text{если } a \leq b \\ b, & \text{в противном случае} \end{cases}$
4	Логика G_{43}	$a \rightarrow b$ G_{43}	$a \rightarrow b = \begin{cases} \min(1, \frac{b}{a}), & \text{при } a \neq 0 \\ 1, & \text{при } a = 0 \end{cases}$
5	L-логика (логи- ка Лукасевича)	$a \rightarrow b$ L	$a \rightarrow b = \min(1, 1 - a + b)$
6	KD-логика	$a \rightarrow b$ KD	$a \rightarrow b = (1 - a) \vee b = \max(1 - a, b)$
7	Z ₁ -логика	$a \rightarrow b$ Z_1	$a \rightarrow b = \begin{cases} 1 - a, & \text{если } a < b \\ 1, & \text{если } a = b \\ b, & \text{если } a > b \end{cases}$
8	Z ₂ -логика	$a \rightarrow b$ Z_2	$a \rightarrow b = \begin{cases} 1, & \text{если } a \leq b \\ 1 - a, & \text{если } a > b \end{cases}$
9	Z ₃ -логика	$a \rightarrow b$ Z_3	$a \rightarrow b = \begin{cases} 1, & \text{если } a \leq b \\ \frac{b}{a + (1 - b)}, & \text{если } a > b \end{cases}$

В. В остальных моделях 1—3 импликации существуют свои определенные преимущества [51], но не учитываются ситуации, которые не могут быть формализованы с помощью набора правил.

В [40] аксиоматический подход к операции «импликация» был подробно описан, постулирующий свойства импликации на интуитивном уровне представления о семантическом содержании нечеткого вывода. Мы не будем подробно описывать здесь аксиомы, а отметим только, что ряд логик табл. 1.4. им отвечает. Так, отношение импликации по правилу логики Лукасевича полностью отвечает аксиоматике [40].

Подробный анализ логического вывода «если A то B » в рамках обобщенных принципов «modus ponens» и modus tol-

Усиление (ослабление) логического вывода в операторе импликации может быть достигнуто путем применения принципа изменения аксиом следующим образом. Если дан нечеткий оператор импликации $A \rightarrow B$ и нечеткое множество B из универсума U , то нечеткое «сильное» множество A из B задается функцией принадлежности $\mu_{\Phi_B}(A)$ в виде $\mu_{\Phi_B}(A) = \bigwedge_{x \in U} (\mu_A(x) \rightarrow \mu_B(x))$. Тогда степень π , с которой A является подмножеством B , есть $\pi(A \subseteq B) = \mu_{\Phi_B}(A)$.

Если нечеткий оператор импликации задан на замкнутом единичном интервале $[0, 1]$, то имеют место следующие соотношения: $a \leftarrow b = b \rightarrow a$ и $a \leftrightarrow b = (a \rightarrow b) \cap (b \rightarrow a)$. Степень, с которой нечеткие множества A и B эквивалентны (или степень их «эквивалентности»), имеет вид: $\pi(A \equiv B) = \pi(A \leq B) \cap \pi(A \geq B)$; $\pi(A \equiv B) = \bigwedge_{x \in X} (\mu_A(x) \leftrightarrow \mu_B(x))$. Напом-

ним, что в теории нечетких множеств операция пересечения \cap и взятия \min эквивалентны. Следует отметить, что в [40] было показано, что в большинстве случаев для практических применений целесообразно работать с многозначными логиками, в которых логические переменные принимают значения из действительного интервала $I = [0, 1]$, разбивая его на 10 подынтервалов, т. е. используя множество $V_1 = \{0; 0,1; 0,2, \dots, 0,9; 1\}$.

Таким образом, операцию «импликация» можно интерпретировать дополнительно как логическую операцию в том или

ином смысле выполнения операции $*$ над соответствующими функциями принадлежности μ_A и μ_B , тем самым усиливая или ослабляя требования к логическому выводу импликация \rightarrow : «если A то B ».

В табл. 1.4 представлены некоторые типовые импликации в многозначных логиках [29, 50].

Приведенные в табл. 1.4 операции дают возможность гибкого применения методов анализа нечеткого вывода для разработки алгоритмов управления нечеткими регуляторами [19, 51]. Было проведено, в частности, сравнение четырех типов импликации, которые использовались при управлении работой парового котла [51] с помощью следующих лингвистических правил:

- $\mu(u, v) = \min(\mu_A(u), \mu_B(v)) = \mu_A(u) \wedge \mu_B(v), u \in U, v \in V;$
- $\mu(u, v) = \min(\mu_A(u), \mu_B(v)) \vee \min((1 - \mu_A(u)), 1);$
- $\mu(u, v) = \min(1, (1 - \mu_A(u) + \mu_B(v))), u \in U, v \in V;$
- $\mu(u, v) = \max((1 - \mu_A(u)), \mu_B(v)), u \in U, v \in V.$

Результаты моделирования показали, что четвертая модель импликации является наиболее удобной и отражает сущность естественного вывода: если посылка A дает следствие B , то посылка A , близкая к A , дает следствие B , также «близкое» к

Таблица 1 5

Вид импликации	Вид правила нечеткого вывода
$\min(\mu_A \cdot \mu_B)$	$R_C = A \times B = \int_{U \times V} \mu_A(u) \wedge \mu_B(v) (u, v)$
$\mu_A(u) \xrightarrow[S]{} \mu_B(v)$	$R_S = A \times V \xrightarrow[S]{} U \times B = \int_{U \times V} [\mu_A(u) \xrightarrow[S]{} \mu_B(v)] (u, v)$
$\mu_A(u) \xrightarrow[G]{} \mu_B(v)$	$R_G = A \times V \xrightarrow[G]{} U \times B = \int_{U \times V} [\mu_A(u) \xrightarrow[G]{} \mu_B(v)] (u, v)$
$\frac{[\mu_A(u) \xrightarrow[S]{} \mu_B(v)]}{\wedge [1 - \mu_A(u) \xrightarrow[G]{} 1 - \mu_B(v)]}$	$R_{SG} = (A \times V \xrightarrow[S]{} U \times B) \cap (\neg A \times V \xrightarrow[G]{} U \times \neg B) =$ $= \int_{U \times V} [\mu_A(u) \xrightarrow[S]{} \mu_B(v)] \wedge [1 - \mu_A(u) \xrightarrow[G]{} 1 - \mu_B(v)] (u, v)$

rens был проведен в [19, 29]. Поскольку импликация с таким нечетким выводом является формирователем базы знаний нечеткого регулятора (см. раздел 2), то такой анализ имеет дополнительный смысл. Приведем вид импликаций и вид нечеткого вывода, которые удовлетворительны с точки зрения обобщенных принципов modus ponens и modus tollens (см. табл. 1.5).

В табл. 1.5 использованы обозначения из табл. 1.4. Таким образом, логика Лукасевича, стандартная последовательность и геделианская последовательность вполне удовлетворительно могут быть использованы для формирования базы знаний нечеткого регулятора. Данный вывод может быть уточнен с учетом результатов [51].

При разработке модели нечетких регуляторов мы будем существенно использовать также нечеткие логики $Z_1 - Z_3$ (см. позицию 7—9 из табл. 1.4).

Проиллюстрируем свойства нечетких логик $Z_1 - Z_3$ на примере аналитических выражений для определения операции двойной импликации (эквивалентности):

$$a \leftrightarrow_{z_1} b = \begin{cases} 1 - ka, & \text{если } a < b, \\ 1, & \text{если } a = b; \\ 1 - kb, & \text{если } a > b, \end{cases}$$

$$a \leftrightarrow_{z_2} b = \begin{cases} 1, & \text{если } a = b, \\ 0, & \text{если } (ka = 1) \vee (kb = 1), \\ \min[(\neg a \cap b), (\neg b \cap a)], & \text{в остальных случаях} \end{cases}$$

$$a \leftrightarrow_{z_3} b = \begin{cases} 1, & \text{если } a = b, \\ 0, & \text{если } (ka = 1) \vee (kb = 1), \\ \min\left(\frac{a}{b + \neg a}, \frac{b}{a + \neg b}\right), & \text{в остальных случаях.} \end{cases}$$

Здесь $ka = \max(a, 1 - a)$ — определение четкости некоторого $a \in V$. Тогда четкость нечеткого множества B определяется как $kB = \bigwedge k\mu_B(x)$. Верхом нечеткого множества B называется $hB = \bigvee_U \mu_B^U(x)$, низом — $\rho B = \bigwedge_U \mu_B(x)$.

Сочетание понятий четкости, нижней и верхней границ нечеткого множества в определении операции импликации в том или ином смысле (из табл. 1.4 или табл. 1.5) дают возможность их гибкого использования в формировании базы знаний нечеткого регулятора и нечетких инструкций соответствующего алгоритма управления (см. раздел 2).

Предложение 1. (Степень возможности включения множеств). Определим функции $\pi_i(A \subseteq B)$ в нечетких логиках $Z_1 - Z_3$ в виде

- $\pi_1(A \subseteq B) = \begin{cases} 1 - \mu_A(x), & \text{если } \mu_A(x) < \mu_B(x), \\ 1, & \text{если } \mu_A(x) = \mu_B(x) \\ \mu_B(x), & \text{если } \mu_A > \mu_B; \end{cases}$
- $\pi_2(A \subseteq B) = \begin{cases} 1, & \text{если } \mu_A(x) \leq \mu_B(x), \\ (1 - \mu_A(x)) \wedge \mu_B(x), & \text{если } \mu_A(x) > \mu_B(x); \end{cases}$
- $\pi_3(A \subseteq B) = \begin{cases} 1, & \text{если } \mu_A(x) \leq \mu_B(x), \\ \frac{\mu_B(x)}{\mu_A(x) + (1 - \mu_B(x))}, & \text{если } \mu_A(x) > \mu_B(x). \end{cases}$

Отметим, что для нечеткой логики Z_1 четкое включение возможно, если $\mu_A(x) = 0$, или $A = \emptyset$. Далее рассмотрим вопрос эквивалентности множеств.

Предложение 2. (Степень возможности эквивалентности множеств). Определим функции $\pi_i(A \equiv B)$ в нечетких логиках $Z_1 - Z_3$ в виде

- $\pi_1(A \equiv B) = \begin{cases} 1 - [(1 - \mu_A(x)) \vee \mu_A(x)], & \text{если } \mu_A(x) < \mu_B(x), \\ 1, & \text{если } \mu_A(x) = \mu_B(x), \\ 1 - [(1 - \mu_B(x)) \vee \mu_B(x)], & \text{если } \mu_A(x) > \mu_B(x). \end{cases}$

$$2. \pi_2(A \equiv B) = \begin{cases} 1, & \text{если } A = B, \\ \bigwedge_T [(1 - \mu_A(x)) \wedge \mu_B(x)] \vee [(1 - \mu_B(x)) \wedge \mu_A(x)], & \text{если } A \neq B, \\ 0, & \text{если } \exists x \parallel \mu_A(x) = 0, \mu_B(x) \neq 0 \text{ (или наоборот)}, \\ a \text{ также } \exists x \parallel \mu_A(x) = 1, \mu_B(x) \neq 1 \text{ (или наоборот)}. \end{cases}$$

$$3. \pi_3(A \equiv B) = \begin{cases} 1, & \text{если } A = B, \\ 0, & \text{если } \exists x \parallel \mu_A(x) = 0, \text{ но } \mu_B(x) \neq 0 \text{ (или наоборот)}, \\ \bigwedge_T \left[\frac{\mu_A(x)}{\mu_B(x) + (1 - \mu_A(x))}, \frac{\mu_B(x)}{\mu_A(x) + (1 - \mu_B(x))} \right], & \text{в противном случае.} \end{cases}$$

В приведенных выражениях символ $|||$ читается «такой, как», множество $T = \{x \in U \mid \mu_A(x) \neq \mu_B(x)\}$, $A = B$ означает, что $\forall x, \mu_A(x) = \mu_B(x)$ или $T = \emptyset$.

Из выражения $\pi_i(A \equiv B)$ следует, что для всех трех нечетких логик $Z_1 - Z_3$ возможность эквивалентности $\pi_i(A \equiv B) = 1$ имеет место только при действительной эквивалентности множеств, т. е. $A = B$. Очевиден также и тот факт, что возможность эквивалентности равна нулю в тех случаях, когда одно из высказываний четко (т. е. либо истинно, либо ложно), а другое нечетко.

Предложение 3. (Степень, с которой нечеткое множество B пусто). Определим операции $\pi_i(B \equiv \emptyset)$ для нечетких логик $Z_1 - Z_3$ в виде

1. $\pi_1(B \equiv \emptyset) = \begin{cases} 1, & \text{если } B = \emptyset, \\ 0, & \text{в противном случае;} \end{cases}$
2. $\pi_2(B \equiv \emptyset) = \begin{cases} 1, & \text{если } hB < 1 \text{ или } B = \emptyset, \\ 0, & \text{в противном случае;} \end{cases}$
3. $\pi_3(B \equiv \emptyset) = \begin{cases} 1, & \text{если } B = \emptyset, \\ 0, & \text{в противном случае.} \end{cases}$

Здесь $B = \emptyset$ означает, что $\forall x, \mu_B(x) = 0$, или эквивалентно $hB = 0$.

Введем теперь понятие «несовместимости» (Disjointness) нечетких множеств. Существует два вида «несовместимости»: первый определяется степенью, с которой одно множество A является подмножеством дополнения второго B , второй — степенью, с которой пересечение множеств пусто.

Предложение 4. Степень несовместимости множеств A и B есть степень, с которой A и B несовместимы:
 1. $\pi(A \text{ disj}_1 B) = \pi(A \subseteq B^c) \wedge \pi(B \subseteq A^c)$; 2. $\pi(A \text{ disj}_2 B) = \pi((A \cap B) = \emptyset)$.

Рассмотрим несовместимость первого вида (при этом примем $T = \{x \mid \mu_A(x) > 1 - \mu_B(x)\}$). Следовательно, имеем

1. $\pi_1(A \text{ disj}_1 B) = \begin{cases} 1, & \text{если } \exists x \ ||| \mu_A(x) = 1 - \mu_B(x), \\ 0, & \text{никогда,} \\ (1 - \mu_A(x)) \wedge (1 - \mu_B(x)), & \text{в остальных случаях;} \end{cases}$
2. $\pi_2(A \text{ disj}_1 B) = \begin{cases} 1, & \text{если } \mu_A(x) \leq 1 - \mu_B(x), \\ 0, & \text{если } \exists x \ ||| \mu_A(x) = 1, \text{ но } \mu_B(x) \neq 0, \mu_B(x) = 1, \\ \text{но } \mu_A(x) \neq 0, \\ \bigwedge_T [(1 - \mu_A(x)), (1 - \mu_B(x))], & \text{в остальных случаях.} \end{cases}$

$$3. \pi_3(A \text{ disj}_1 B) = \begin{cases} 1, & \text{если } \mu_A(x) = \mu_B(x) \text{ или } \mu_B(x) = 0, \\ 0, & \text{никогда} \\ \bigwedge_T \left[\frac{1 - \mu_A(x)}{\mu_A(x) + (1 - \mu_B(x))}, \frac{1 - \mu_B(x)}{\mu_B(x) + (1 - \mu_A(x))} \right], & \text{в противном случае.} \end{cases}$$

Отметим, что степень несовместимости множеств равна нулю только для нечеткой логики Z_2 . При этом обязательным условием является нормальность одного из рассматриваемых нечетких множеств при одновременной субнормальности другого.

Предложение 5. (Степень, с которой множество является подмножеством своего дополнения). Для рассматриваемых нечетких логик $\pi_i(A \subseteq A^c)$ будет иметь вид

1. $\pi_1(A \subseteq A^c) = \begin{cases} 1, & \text{если } hA = 0, \\ 0, & \text{если } hA = 1, \\ 1 - hA, & \text{в остальных случаях;} \end{cases}$
2. $\pi_2(A \subseteq A^c) = \begin{cases} 1, & \text{если } hA \leq 0,5, \\ 0, & \text{если } hA = 1, \\ 1 - hA, & \text{в остальных случаях;} \end{cases}$
3. $\pi_3(A \subseteq A^c) = \begin{cases} 1, & \text{если } hA \leq 0,5, \\ 0, & \text{если } hA = 1, \\ \frac{1 - hA}{2hA}, & \text{в остальных случаях.} \end{cases}$

Очевидно, что для нечеткой логики Z_1 степень, с которой множество является подмножеством своего дополнения, равна степени, с которой это множество является пустым. Следует также отметить, что семантический анализ, проведенный в [52], а также приведенный анализ показывают большую схожесть в свойствах нечеткой логики Z_1 и KD -логики (см. позицию 6 из табл. 1.4). Однако, как будет показано в разделе 2, нечеткая логика Z_1 имеет ряд преимуществ по сравнению с KD -логикой, позволяющих успешно ее использовать при формализации улучшенных правил условного нечеткого вывода, а также для моделирования различных технологических процессов.

Семантический анализ операции импликации в многозначных логиках оказался очень важным при исследовании процессов измерений и решения возможных парадоксов в квантовой механике [47, 48, 53—58]. Так, в [54] показано, что стандартная модель квантовой (нерелятивистской) механики содержит множество неразличимых различных состояний частицы и такого рода избыточность устраняется после введения в модель квантовой механики нечетких (совместных) наблюдаемых. При этом размытость состояний уже не может интерпретироваться как субъективность наблюдения, а характеризуется объективностью исследуемого физического процесса [55].

На конкретных примерах исследования процессов измерения координаты и импульса квантовой частицы в [55, 57, 58] показана возможность введения нечетких наблюдаемых и установлена непротиворечивость интерпретации состояния индивидуальной частицы как нечеткой точки $((q, f_q), (p, g_p))$ с нечеткими функциями распределения f_q и g_p соответственно относительно координаты q и импульса p частицы, а соотношение неопределенности имеет аналогичный определению Гейзенберга вид $\Delta_f Q \Delta_g D \geq \hbar/2$. Это соотношение устанавливает нижнюю границу (индивидуальной) размытости результатов измерений и непосредственно связано с совместными измерениями.

Подобного рода результаты имеют важное значение при исследовании квантовых систем как моделей объектов управления [53, 59, 60]. В [60] показаны возможные подходы к построению квантовых систем как моделей с нечеткими параметрами в структуре (уравнение Шредингера с нечетким параметром) и дана интерпретация нового соотношения неопределенности нечетких измерений. Физические аспекты и обсуждения особенностей модели квантовой механики с нечеткими параметрами [60] читатель может найти в [61].

В моделях рассуждений с нечеткой информацией, использующих отношение моделирования, с помощью нечеткого отображения $R: W \rightarrow M$ (где W, M — пространства состояний внешней среды и модели соответственно) осуществляется отображение в нечеткую универсальную шкалу, инвариантную для ряда однородных семантических ситуаций [13, 19]. При этом алгебра расширенных операций над функциями принадлежности лингвистических переменных в пространстве состояний W изоморфна (подобна) алгебре расширенных функций в пространстве состояний модели M [19]. Для задания некоторой псевдофизической логики рассуждений такой алгебры бывает часто достаточно. В [19, 62] приведены примеры подобного рода логики. На примере ориентации на местности, что эквивалентно соответствующим операциям в неметрической пространственной логике [62], рассмотрены особенности и возможности такого подхода.

Возникающие в физических моделях квантовой механики парадоксы привели к необходимости расширения логики рассуждений, не ограничиваясь только пространственными отображениями. Так, квантовая логика для взаимодействующих частиц [56, 63] для снятия парадокса квантовых измерений по необходимости должна быть (помимо особенностей квантового языка рассуждений) дополнена структурой языка принципа относительности, что приводит к квантовой релятивистской логике [56]. При этом импликация «если... то», используемая в логических выводах наблюдателей объективных и субъективных состояний и измерений, рассматриваются как пространственно-временные отношения (с точки зрения соответствующих наблюдателей в заданных системах отсчета).

Такой подход привел в [64] к прямому применению преобразования Лоренца пространственно-временного континуума к нечетким множествам, названным релятивистскими нечеткими множествами [53, 64], и обобщению понятия «пространственно-временного континуума» как «пространство-время» с нечеткими событиями [65, 66]. Так, в [64] под релятивистским нечетким множеством A понимается следующее определение. Пусть $A \subset X = \{x\}$ — заданное множество и пусть $\mu_A(x)$ и $\mu_{\bar{A}}(x)$ — функции принадлежности A и его дополнения \bar{A} . Наблюдатель R' , который в заданной системе отсчета S наблюдает события множества A ($x \in A$ или $x \notin A$), характеризуется релятивистскими функциями принадлежности $\mu_A(x|R')$ и $\mu_{\bar{A}}(x|R')$ следующего вида:

$$\mu'_A(x|R') = \rho(u_A) [\mu_A(x) + u_A(x|R') \mu_{\bar{A}}(x)],$$

$$\mu'_{\bar{A}}(x|R') = \rho(u_A) [\mu_{\bar{A}}(x) + u_A(x|R') \mu_A(x)],$$

где отображение $u_A: E \rightarrow [0, M] \subset [0, 1]$, E — пространство объектов (точечных событий), $A \subset E$, $x \rightarrow u(x|R') \geq 0$, функция, характеризующая нечеткость наблюдения A наблюдателем R' и $\rho(u_A \triangleq [1 - u^2(R|R')/C^2]^{-1/2})$. При этом запись $u(R|R')$ означает, что процесс наблюдения R' осуществляется через другого наблюдателя R и поэтому характеризует нечеткость наблюдения. С точки зрения специальной теории относительности функция $u_A(R|R')$ является аналогом относительной скорости «движения» наблюдателя R' относительно наблюдателя R в системе отсчета S . Таким образом, можно построить нечеткое релятивистское исчисление с соответствующими операциями объединения, пересечения и т. п. над нечеткими релятивистскими множествами [64].

Использование идей нечеткой логики, как отмечено в [67], может оказать существенное влияние и на построение архитектуры средств ВТ и ЭВМ, так как могут снизить предельные ограничения на процессы обработки информации при заданных физических ограничениях, полученных из квантовой механики и термодинамики. При этом сами физические ограничения могут быть несущественными, если речь идет о построении процессоров для выполнения только логических операций [68]. Конкретные примеры реализации логических связей нечеткой логики на БИС для нечетких микроконтроллеров можно найти в [69].

В заключение отметим, что нечеткая логика является основой для построения нечетких алгоритмов управления, когда импликация вида «если A то B » и «если A то B иначе C » в качестве A , B и C содержат соответствующие нечеткие инструкции, описываемые лингвистическими переменными [6, 2, 15—19]. В следующем разделе 2 на конкретных примерах построения нечетких регуляторов рассмотрены вопросы построения нечетких

алгоритмов управления и применения моделей нечеткой логики для лингвистической аппроксимации описания объектов управления.

2. НЕЧЕТКИЕ МОДЕЛИ ДИНАМИЧЕСКИХ СИСТЕМ УПРАВЛЕНИЯ

В данном разделе основное внимание уделено качественному описанию принципов построения нечетких регуляторов, широко применяемых при разработке робототехнических комплексов, АСУТП, гибких производственных систем, систем управления сложными динамическими системами и др. Изложение материала построено так, чтобы максимально приблизить исследуемые модели к традиционным объектам теории систем управления. Это дает возможность осветить особенности теории нечетких моделей систем управления и показать ряд преимуществ нечетких регуляторов.

2.1. Принципы построения нечетких регуляторов

На практике во многих случаях автоматическое управление технологическими процессами происходит в условиях неопределенности, связанной с отсутствием достаточной статистики о поведении управляемых объектов. Проектировщикам систем управления такими объектами приходится также учитывать некоторые неформализуемые или трудноформализуемые факторы. Уровень сложности подобных систем настолько высок, что использование известных детерминированных и стохастических моделей для их проектирования не обеспечивает желаемых характеристик. В этих случаях адекватные математические модели управляемых систем могут основываться на теории нечетких множеств, позволяющих синтезировать интеллектуальные системы управления.

Основу проектирования интеллектуальных нечетких регуляторов составляет конструирование «знаний» с использованием методов представления и поиска знаний. Поэтому создание нечетких промышленных регуляторов базируется на принципах бурно развивающейся в последнее время теории искусственного интеллекта.

Пример 1. Структура интеллектуальной системы управления с нечетким регулятором представлена на рис. 1. Выходная переменная объекта регулирования — технологического процесса y сравнивается с ее заданным значением g , и ошибка рассогласования ε поступает как в масштабный элемент 3 с коэффициентом k_ε , так и в дифференциатор u , выход которого умножается на k_ε в масштабном элементе 5. Элементы 6, 7 предназначены для преобразования текущих значений рассогласования



Рис. 1. Структурная схема интеллектуальной системы управления с нечетким регулятором

ε и производной от рассогласования в их лингвистические значения. Нечеткие значения ε^* , ε^{**} поступают в главный элемент нечеткого регулятора — базу знаний (БЗ). Как правило, БЗ нечетких регуляторов строится на основе продукционной модели знаний, имеющей конструкцию вида «если..., то...». Каждая продукция, представляющая собой множество пар ситуация — действие, позволяет ставить в соответствии со сложившейся ситуацией действие регулятора в виде значения регулирующего воздействия на объект.

Найденное лингвистическое значение управления после умножения на масштабный коэффициент k_u в элементе 8 и преобразования в его четкое значение u поступает на исполнительный элемент объекта управления.

При синтезе нечетких регуляторов основной проблемой является конструирование его БЗ. В БЗ опыт и знание человека-оператора (эксперта) можно вносить следующими способами [71]: 1) оператор-эксперт управляет технологическим процессом, за которым «наблюдает» регулятор; последний запоминает все действия эксперта и заполняет свою БЗ; 2) оператор-эксперт формирует свое действие при каждой наблюдаемой ситуации в виде продукции «если..., то...», множество которых составляет БЗ регулятора; 3) перед самоорганизующимся нечетким регулятором ставится цель обеспечить желаемую переходную характеристику проектируемой промышленной системы управления; одновременно сообщается некоторая информация о технологическом процессе — объекте управления. Регулятор самостоятельно методом проб и ошибок накапливает знания без эксперта.

Естественно, что в качестве оператора-эксперта выступает квалифицированный специалист.

Здесь следует отметить, что формирование причинно-следственных связей для БЗ может осуществляться не только на основе продукционных правил. Типичным примером является формирование БЗ для медицинских экспертных систем.

В общем случае представление знаний в управляющих системах в теории искусственного интеллекта осуществляется логическими, реляционными, фреймовыми и продукционными языками [43, 13]. Учитывая такую важную специфику работы систем автоматического регулирования, как реальный режим работы (on-line) и удобство представления информации о процедурах и условиях их применения, в дальнейшем в этом разделе будем использовать продукционную модель описания знаний в нечетких регуляторах. Другие варианты рассмотрены при обсуждении экспертных медицинских систем [44].

Каждая продукция дается в виде набора правил, представляющих собой фрагмент знаний — ядро в инженерии знаний, и имеет вид «условие — действие».

Левая часть каждой продукции рассматривается как конъюнкция элементарных (перцепционных) условий, а правая часть — как множество элементарных действий. Для регулятора, представленного на рис. 1, любое правило в базе знаний может быть представлено как

если (ϵ есть ϵ_1^*) и ($\dot{\epsilon}$ есть $\dot{\epsilon}_1^*$), тогда (u есть u_1^*),

где ϵ , $\dot{\epsilon}$, u — переменные, а ϵ_1^* , $\dot{\epsilon}_1^*$, u_1^* — их лингвистические значения.

В табл. 2.1 представлены как пример лингвистические правила, являющиеся БЗ нечеткого промышленного регулятора для управления ректификационной колонной. При этом использова-

ны следующие обозначения лингвистических переменных: ОБ — отрицательно большое; ОНБ — отрицательно ниже большого; ОС — отрицательно среднее; ОНС — отрицательно ниже среднего; ОМ — отрицательно маленькое; ОНМ — отрицательно ниже маленького; ОН — отрицательно нуль; ПН — положительно нуль; ПНМ — положительно ниже маленького; ПМ — положительно маленькое; ПНС — положительно ниже среднего; ПС — положительно среднее; ПНБ — положительно ниже большого; ПБ — положительно большое.

Масштабные коэффициенты k_ϵ , $k_{\dot{\epsilon}}$, k_u являются элементами универсальных множеств E , E' и U (на которых определяются нечеткие множества ϵ^* , $\dot{\epsilon}^*$ и u^*) и определяются исходя из условий конкретного управляемого объекта. Например, если универсальное множество E является $(-10, -9, \dots, +9, +10)$ и требуется, чтобы параметр рассогласования ϵ в системе находился в диапазоне $(-1, +1)$, то тогда k_ϵ берется равным 10 для того, чтобы нечеткий регулятор мог использовать все универсальное множество, на котором определяются нечеткие множества.

Применение теории нечетких множеств в проектировании регуляторов позволяет повышать их «интеллект», компетентность, приблизив к интеллекту человека. «Очеловечивание» нечетких регуляторов является одной из центральных проблем в современной теории и технике автоматического регулирования. Дадим краткий исторический обзор работ этого направления и укажем некоторые общие моменты и различия отмеченных исследований, здесь выбраны только наиболее типичные работы.

В одной из первых работ [70], посвященных практическому применению нечетких логических регуляторов (НЛР), описан эксперимент с эвристическим синтезом регулятора для управления паровой турбиной. Сравнительный анализ результатов регулирования стандартными средствами в виде нелинейных цифровых устройств и нечетким регулятором показал преимущество последнего. Аналогичный анализ функционирования ПИ-регулятора и НЛР, используемых в системе управления нагревом воды, приведен в [2, 19]. Необходимо отметить ряд общих особенностей, проявившихся при испытаниях НЛР:

— при исследовании и проектировании НЛР авторы не опирались на точную модель процесса (при этом интуиция разработчика, его знания о процессе непосредственно учитывались в алгоритме управления; далее следовал итерационный процесс, заключающийся в проверке функционирования алгоритма, изучении его поведения и последующей модификации соответствующих управляющих правил; часто эта процедура требует значительных затрат времени);

— нечеткие подмножества включали в себя аналогичные лингвистические переменные, например ПБ (положительно

Таблица 2.1

		$\dot{\epsilon}$													
		ОБ	ОНБ	ОС	ОНС	ОМ	ОНМ	ОН	ПНМ	ПМ	ПНС	ПС	ПНБ	ПБ	
ϵ	ОБ	НБ	ПБ	НБ	ПБ	ПБ	ПБ	ПБ	ПБ	ПБ	ПБ	ПБ	ПБ	ПБ	
	ОНБ	ПНБ	ПНБ	ПНБ	ПНБ	ПНБ	ПНБ	ПНБ	ПНБ	ПНБ	ПНБ	ПНБ	ПНБ	ПНБ	
	ОС	ПНБ	ПНБ	ПНБ	ПНБ	ПНБ	ПС	ПС	ПС	ПНС	ПС	ПНБ	ПБ	ПБ	
	ОНС	ПНБ	ПНБ	ПНБ	ПНБ	ПС	ПНС	ПНС	ПНС	ПС	ПНБ	ПНБ	ПБ	ПБ	
	ОМ	ПНБ	ПНБ	ПНБ	ПС	ПНС	ПМ	ПМ	НС	ПНС	ПС	ПНБ	ПБ	ПБ	
	ОНМ	ПНБ	ПНБ	ПНБ	ПНС	ПМ	ПНС	ПМ	ПНС	ПНС	ПС	ПНБ	ПБ	ПБ	
	ОН	ПНБ	ПНБ	ПС	ПНС	ПНМ	ПМ	ПМ	ПНМ	ПНС	ПБ	ПНБ	ПНБ	ПБ	
	ПМ	ОНБ	ОНБ	ОС	ПНМ	ПМ	ОН	ОН	ОН	ОНС	ОС	ОНБ	ОБ	ОБ	
	ПНМ	ОНБ	ОНБ	ОС	ОНС	ОНМ	ОН	ОН	ОН	ОНС	ОС	ОНБ	ОБ	ОБ	
	ПМ	ОНБ	ОНБ	ОС	ОНС	ОМ	ОН	ОМ	ОНМ	ОС	ОНБ	ОБ	ОБ	ОБ	
	ПНС	ОБ	ОНБ	ОС	ОНС	ОНС	ОМ	ОНС	ОНС	ОНБ	ОБ	ОБ	ОБ	ОБ	
	ПС	ОБ	ОНБ	ОНБ	ОНБ	ОС	ОС	ОНС	ОС	ОС	ОБ	ОБ	ОБ	ОБ	
	ПНБ	ОБ	ОБ	ОНБ	ОНБ	ОНБ	ОС	ОНБ	ОНБ	ОБ	ОБ	ОБ	ОБ	ОБ	
	ПБ	ОБ	ОБ	ОБ	ОБ	ОБ	ОБ	ОБ	ОБ	ОБ	ОБ	ОБ	ОБ	ОБ	

и

большой) или ОМ (отрицательно малый), что отражает общий подход к выбору определенных интервалов квантованных величин;

— как правило, во всех НЛР использовался основополагающий принцип регулирования — принцип регулирования по отклонению.

Наряду с общими моментами в подходе к проблеме создания НЛР существуют и различия как в выборе структуры регулятора, так и в его применении. К основным различиям относится выбор единственного управляющего воздействия из выходного нечеткого подмножества. Один из применяемых способов — выбор такого значения управления, для которого функция принадлежности имеет наибольшее значение (при наличии нескольких точек с максимальной функцией принадлежности выбирается среднее значение для этих точек). Это так называемый метод *mean of maxima* (среднего по максимуму). Другой способ — выбор такого значения управления, которое является медианой фигуры, ограниченной кривой функции принадлежности выходного нечеткого подмножества. Такой способ получил название *centre of area*, (метод центра площади). Существуют и другие возможные подходы. Так, при использовании метода *mean of maxima* нечеткий алгоритм ведет себя как позиционное реле и для анализа НЛР применимы методы классической теории нелинейных САУ [71]. В то же время при использовании метода центра площади нечеткий алгоритм идентичен ПИ-регулятору, и, согласно [71], считается, что этот метод предпочтителен. Однако наиболее интересным моментом [71] является методика синтеза НЛР на основе лингвистической модели объекта, которая демонстрируется на ряде примеров. Но авторы ограничились примерами для объектов первого порядка, не указав способы и возможность распространения этой методики на объекты более высоких порядков, что существенно ограничивает ее применение (см. далее подраздел 2.3).

Различия имеются и в вопросе выбора входов для НЛР. В качестве входов для НЛР рассматривались либо ошибка и скорость изменения ошибки, либо ошибка и суммарная ошибка. Обсуждался алгоритм, где входами были ошибка и скорость изменения ошибки, а выход определялся двумя способами: если ошибка была велика, то алгоритм выдавал абсолютное значение управления, если же ошибка была мала, то алгоритм выдавал приращение управляющего воздействия. Такой подход применялся с целью оптимизации времени отклика системы.

Необходимо отметить, что во всех случаях выдвигались требования соответствия НЛР традиционным критериям: качеству переходного процесса и декомпозиции каналов регулирования.

Для применения НЛР использовался широкий спектр малых и больших ЭВМ и языков программирования типа ФОРТРАН, АПЛ, БЕЙСИК, язык ассемблера.

Необходимо также отметить ряд работ, внесших вклад в развитие теории моделей нечетких САУ. Так, в [72] рассматривается концепция «оператора наблюдения», с помощью которого можно точнее определить состояние процесса. Была также предложена идея «цели регулирования».

Постановка задачи в этом случае заключается в следующем. Пусть G — нечеткое множество в пространстве состояний X . Тогда описание процессов включает в себя: отображение в пространстве состояний $f: X \times U \rightarrow X$, оператор наблюдения $Q: x \circ Q = y$ и отображение для регулятора $g: Y \rightarrow U$.

Значительные результаты получены в [73], где были заложены математические основы анализа нечетких систем. Так, например, были определены отображения $f: F(x) \times F(u) \rightarrow F(x)$ и $g: F(x) \rightarrow F(y)$, где $F(x)$, $F(y)$ и $F(u)$ — множества всех нечетких подмножеств пространства состояний X , входного пространства U и выходного пространства Y . Также были введены понятия достижимости, наблюдаемости и устойчивости, которые обобщают концепции четких систем.

В [74] была сформулирована теорема об аппроксимации лингвистических алгоритмов нечетких систем аналитическими функциями. Указанная теорема заложила основы аналитической теории нечетких САУ.

Отметим также [75], в которой введены арифметические операции на нечетких множествах и на их базе определены нечеткий интеграл свертки и нечеткая передаточная функция. Однако в методике появлялось огромное количество избыточной информации и возникали трудности с практической реализацией.

Проблемы теории нечетких систем обсуждаются в [76]. Рассматриваются вопросы устойчивости нечетких систем и для них формулируется принцип инвариантности.

Результаты разработки нечеткой САУ для нелинейного многомерного процесса представлены в [77]. Обсуждается вопрос декомпозиции объекта управления (паровой турбины) для того, чтобы подсистемы обладали асимптотической устойчивостью. В результате моделирования было показано, что замкнутые подсистемы отслеживают желаемые параметры состояния при различных условиях работы объекта управления. Проведен анализ устойчивости с использованием частотного критерия в предположении, что декомпозированные системы линейны и декомпозиция проведена корректно. В заключение высказаны предположения, какой тип вариации установки и структуры регулятора является допустимым.

Применение нечеткой логики в обучающихся регуляторах обсуждается в [78]. Показывается, что для ряда случаев необходимо перейти от чисто описательного подхода к предписывающему или к самоорганизующейся системе. Недостатком нечеткой логики, как указывает автор, является ее описывающая форма представления знаний, что не позволяет применять данный под-

ход для принятия решений. Отмечается, что нечеткая логика, как и любая другая, может определять следствие из предварительно установленных предпосылок. Предписывающая система возможна, если используется иерархический подход к принятию решений; при этом стратегия на нижнем уровне определяется исходя из описания на верхнем уровне. Самоорганизующийся регулятор, отмечается в [78], использует эту идею для определения управляющих правил из установленного желаемого отклика системы и предшествующего изучения поведения объекта управления.

Вообще, как отмечается в [79], перспективным направлением в развитии НЛР является создание адаптивных и самообучающихся нечетких систем. Этим можно объяснить появление ряда публикаций [80, 81], в которых обсуждаются вопросы проектирования самоорганизующихся регуляторов (СОР). Этот тип регуляторов представляет собой иерархическую структуру, где на нижнем уровне находится собственно нечеткий регулятор, а на верхнем уровне — монитор (корректирующее устройство), осуществляющий при необходимости модификацию правил регулятора нижнего уровня. Монитор использует желаемый отклик замкнутой системы, представленный в виде таблицы соответствий. Эта таблица имеет входы, идентичные входам НЛР (т. е. ошибка и скорость изменения ошибки), а элементы таблицы показывают, насколько определенное состояние системы предположительно отличается от желаемого. Таким образом, указанная таблица соответствий отражает те изменения, которые необходимо внести в структуру системы, изучая ее поведение в пространстве состояний. Нулевые элементы в таблице верхнего уровня образуют область в пространстве состояний, в которой характеристика НЛР является удовлетворительной. Для применения СОР предполагается наличие непрерывной зависимости выходов от входов. Если поведение системы отлично от желаемого, то таблица верхнего уровня воздействует на соответствующие элементы таблицы нижнего уровня (поэтому элементы верхней таблицы трактуются как усиления). Вводится также параметр P , называемый «задержкой в усилении». Отмечается итеративный характер действия СОР. Подобный СОР был применен для управления паровым котлом в двух вариантах: в первом случае использовались правила, сформулированные человеком-оператором, во втором — СОР до начала эксперимента не содержал ни одного правила. Результаты обоих вариантов после окончания итеративной процедуры самообучения значительно различались. Это связано с тем, что желаемая реакция системы не задается единственным образом, а лежит в некоторой области, в которой можно определить множество удовлетворительных решений. Отмечается, что СОР может функционировать с первоначальным набором даже неадекватных правил. Особое значение применение СОР имеет в системах управления

вспомогательными аппаратами замещения утраченных функций и искусственных органов (типа аппаратов искусственной вентиляции легких, искусственной почки, искусственного кровообращения и т. д.) [83]. В этом случае применение нечетких алгоритмов в сочетании с принципами квантовой нечеткой логики [48], адаптации и самоорганизации позволяет существенно улучшить процессы управления в экстремальных ситуациях.

В связи с проведенным обзором работ по НЛР и СОР необходимо отметить следующее:

— применение НЛР позволяет использовать для целей управления информацию качественного характера, которую невозможно формализовать при реализации традиционных законов регулирования, при этом НЛР оказываются малочувствительными к возмущениям в определенном диапазоне и отличаются улучшенными характеристиками по сравнению с классическими регуляторами;

— для составления управляющих правил НЛР требовались интуиция разработчика и хорошее знание объекта управления, однако в литературе практически отсутствует какая-либо методика для непосредственного синтеза НЛР;

— изменение параметров объекта управления требует модификации управляющих правил НЛР с их последующей корректировкой, для реализации этой процедуры необходимы большие затраты времени;

— как правило, СОР использовался для первоначального синтеза таблицы нижнего уровня; в рассмотренных работах отсутствует информация о возможности использования СОР в качестве промышленной САУ; исключение составляет [78], где СОР применялся для синтеза управляющих правил при экспериментальном управлении паровым котлом;

— недостаточно освещена возможность использования СОР для многомерного процесса (возможно, это связано с необходимостью снабжения СОР дополнительным блоком, содержащим грубую модель объекта управления); такая модель представляется в виде якобиана, что вызывает дополнительные трудности для его расчета в процессе управления;

— в литературе (как в отечественной, так и зарубежной) отсутствуют методологические аспекты построения верхнего иерархического уровня СОР как задатчика желаемой характеристики системы управления;

— не имеется публикаций о возможности и путях применения СОР для управления нестационарными многомерными промышленными объектами.

В данном разделе рассматриваются возможные подходы к решению некоторых из указанных задач и современное состояние проблемы построения и использования нечетких регуляторов и САУ.

2.2. Аналитические методы синтеза нечетких регуляторов

Рассмотрим методы синтеза систем автоматического управления по критерию абсолютной инвариантности выходной переменной по отношению к внешним возмущениям. Теория инвариантности (разработанная акад. Б. Н. Петровым [84] и его учениками и последователями) тесно связана с идеями компенсации возмущений и каналов обратных связей, в разработке и становлении которых принимал активное участие Г. М. Уланов. Здесь проиллюстрируем отмеченную во введении взаимосвязь методов теории инвариантности (компенсации возмущений) и продукционных правил формирования обратных связей.

Пример 1. Сначала рассмотрим нечеткую систему автоматического регулирования (САР), структурная схема которой представлена на рис. 2. САР функционирует следующим обра-

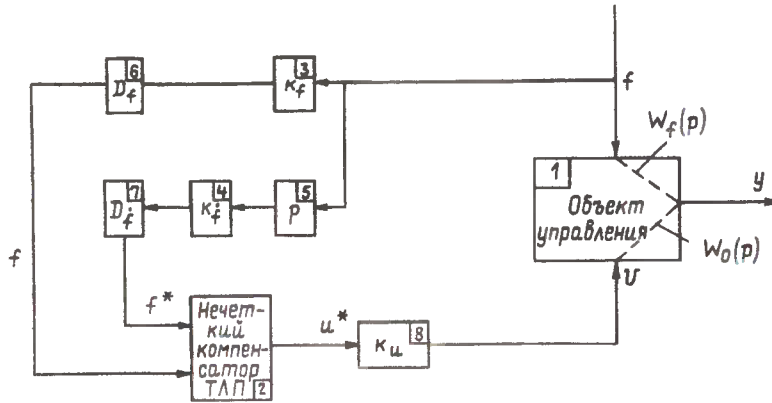


Рис. 2. Структурная схема нечеткой разомкнутой системы управления

зом. Изменяемое возмущение f , проходя по двум каналам через объект регулирования 1, влияет на регулируемый параметр y . Первый канал, являясь естественным, учитывает изменение y в соответствии с передаточной функцией объекта по возмущающему каналу $W_f(p)$. По искусственно организованному второму каналу системы возмущение f проходит следующим образом. Его текущее значение и производная, определяемая элементом 5, после умножения соответственно на коэффициенты k_i и k_i' (элементы 3, 4) и последующего преобразования в элементах 6, 7 в их лингвистические значения, передаются на вход нечеткого компенсатора 2. Выход последнего после умножения на масштабный коэффициент k_u (элемент 8) поступает на вход объекта, изменяя y в соответствии с передаточной функцией $W_0(p)$. Задача синтеза рассматриваемой системы по критерию инвариантности y относительно f заключается в конструирова-

нии такого закона регулирования $u^* = R(f^*, \dot{f}^*)$, который смог бы обеспечить независимость выходного параметра системы y при произвольных изменениях возмущения f , где u^* , f^* , \dot{f}^* — множества значений лингвистических переменных — соответственно выходного параметра регулятора (компенсатора) 2, возмущения и его производной.

Идея описываемого метода синтеза подобных систем заключается в аппроксимации нечетких отношений R функциями в четкой области с дальнейшим использованием аналитических методов их оптимального конструирования.

Как известно из теории автоматического управления, в классических системах управления по возмущению закон регулирования описывает обычное отображение:

$$\Phi_0: F \rightarrow U, \quad (2.1)$$

где F — пространство входов, U — пространство управляющих воздействий.

Если регулятор реализован в классе нечетких отображений типа

$$\begin{aligned} &\text{если } f_1^* \text{ и если } \dot{f}_1^*, \text{ то } u_1^* \text{ и} \\ &\text{если } f_2^* \text{ и если } \dot{f}_2^*, \text{ то } u_2^* \text{ и} \\ &\dots \dots \dots \\ &\text{если } f_n^* \text{ и если } \dot{f}_n^*, \text{ то } u_n^*, \end{aligned} \quad (2.2)$$

то, аналитически аппроксимируя их, можно сконструировать отображение Φ_0 .

Сформулированная ниже теорема утверждает, что всегда можно сконструировать функцию отображения Φ_0 , адекватную по точности нечеткому отображению вида (2.2).

Теорема. Пусть R — нечеткое отображение вида (2.2). Для $\forall \delta$ существует такая аналитическая аппроксимация Φ_0 вида (2.1), при которой $\rho(\Phi_0, R) \leq \delta$, здесь $\rho(\dots)$ — заданная метрика.

Отметим, что аппроксимацию отображений можно строить несколькими способами. Наиболее точной считается аппроксимация, определенная как композиция

$$U = \Phi_0(f) = f_3 \circ [f_2 \circ (f_1(f))], \quad (2.3)$$

где $f_1: F \rightarrow A$, т. е. $\mu = f_1(f)$; $f_2: A \rightarrow B$, т. е. $\beta = f_2(\mu)$; $f_3: B \rightarrow U$, т. е. $u = f_3(\beta)$.

Тогда

$$\Phi_0(f) = f_3 \circ [f_2 \circ (f_1(f))], \quad \Phi_0: F \rightarrow U. \quad (2.4)$$

Здесь $f_1(f)$, $f_2(\mu)$ и $f_3(\beta)$ определяются как решения задачи

$$\begin{aligned} \mu &= f_1(f, a_1^1, a_2^1, \dots, a_n^1); \\ Q_1 &= (a_1^1, a_2^1, \dots, a_n^1) = \sum_{i=1}^n [f_i(f_i, a_1^1, a_2^1, \dots, a_n^1) - \mu_i]^2 \rightarrow \min; \end{aligned}$$

$$\beta = f_2(\mu, a_1^1, a_2^1, \dots, a_n^1);$$

$$Q_2 = (a_1^2, a_2^2, \dots, a_n^2) = \sum_{i=1}^n [f_2(\mu_i, a_1^2, a_2^2, \dots, a_n^2) - \beta_i]^2 \rightarrow \min;$$

$$U = f_3(\beta, a_1^3, a_2^3, \dots, a_n^3);$$

$$Q_3 = (a_1^3, a_2^3, \dots, a_n^3) = \sum_{i=1}^n [f_3(\beta_i, a_1^3, a_2^3, \dots, a_n^3) - U_i]^2 \rightarrow \min.$$

Следует отметить, что такая аппроксимация учитывает все параметры в нечетком множестве.

Однако ввиду сложности формул даже для определения промежуточного отображения этот метод построения аппроксимации мало приемлем для практической реализации. Поэтому другой метод определения аппроксимирующего отображения заключается в определении множества точек $(f_1^*, u_1^*) \dots, (f_n^*, u_n^*)$ в пространстве $F \times U$ с учетом степени принадлежности элементов нечетких множеств A и B соответственно. Здесь из-за громоздкости подробное доказательство и сам алгоритм не приводятся.

Таким образом, отображение Φ_0 , т. е. аналитически определенная функция вида (2.1) нечеткого компенсатора (нечеткого регулятора), позволяет исследовать систему управления (с нечетким регулятором) на устойчивость и показатели качества, в т. ч. инвариантность, с использованием классической теории автоматического управления.

Пример 2. Пусть нечеткий компенсатор определяется таблицей логических высказываний с использованием лингвистических переменных типа (2.2):

если значение возмущения ПОЛОЖИТЕЛЬНО БОЛЬШОЕ, тогда если значение скорости изменения возмущения ОТРИЦАТЕЛЬНО МАЛОЕ, тогда значение управления ПОЛОЖИТЕЛЬНО СРЕДНЕЕ иначе...

если значение возмущения ОТРИЦАТЕЛЬНО БОЛЬШОЕ тогда если значение скорости изменения возмущения ПОЛОЖИТЕЛЬНО БОЛЬШОЕ тогда значение управления ПОЛОЖИТЕЛЬНО СРЕДНЕЕ или...

Приведенные лингвистические переменные и продукционное правило можно записать в виде

$$R_1 = \text{если } f_1^*, \text{ тогда если } f_1^*, \text{ то } u_1^*. \quad (2.6)$$

Нечеткое описание алгоритма компенсатора будет иметь тогда вид:

$$u^* = R(f^*, \dot{f}^*). \quad (2.7)$$

Используя приведенную методику аппроксимации, нечеткий компенсатор, описанный уравнением (2.7), можно представить

следующим аналитическим выражением:

$$u = \Phi_0(f, \dot{f}). \quad (2.8)$$

Эквивалентная (нечеткой) структура разомкнутой системы управления представлена на рис. 3.

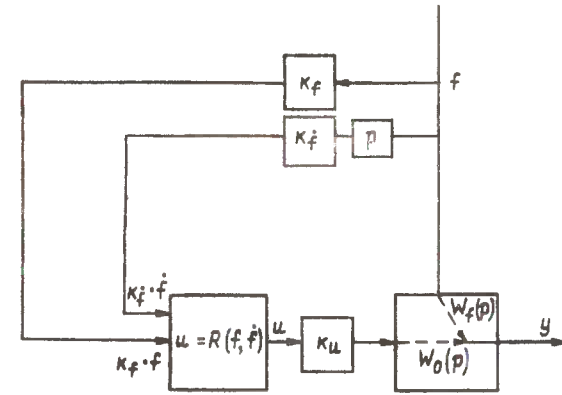


Рис. 3. Структурная схема эквивалентной нечеткой разомкнутой системы управления

Условия инвариантности, согласно структуре рис. 3, определяются как

$$\begin{aligned} W_f(p) f(p) &= k_u W_0(p) u(p), \quad u = R(f, \dot{f}), \\ f &= k_f \cdot f, \quad \dot{f} = k_{\dot{f}} \cdot \dot{f}, \end{aligned} \quad (2.9)$$

где $W_f(p)$, $W_0(p)$ — соответственно передаточные функции по возмущающему и управляющему каналам объекта; k_f и k_u — постоянные коэффициенты

Теперь можно получить условия инвариантности в нечеткой комбинированной системе управления, структура которой представлена на рис. 4. Для этого аппроксимируем нечеткий компенсатор и регулятор (т. е. таблицы лингвистических правил (ТЛП) ТЛП-1 и ТЛП-2, показанные на рис. 4) аналитическими функциями вида:

$$u_j = R(f, \dot{f}), \quad u_e = R(e, \dot{e}). \quad (2.10)$$

Согласно составленной эквивалентной структуре системы (рис. 5), в нелинейной комбинированной системе нелинейный регулятор не включен в разомкнутый контур, включающим в себя нелинейный компенсатор. Следовательно, наличие регулятора не влияет на условие инвариантности в системе и он будет определяться по уравнению (2.9), а u в виде:

$$u = u_j + u_e.$$

значительными затратами времени, но и с технологическими сложностями.

В то же время можно получить качественную информацию о технологическом процессе непосредственно у лица, управляющего объектом. Иными словами, возможно нечеткое описание объекта, которым человек-оператор успешно пользуется при управлении процессом. В связи с этим представляется целесообразным применять нечеткое описание объекта непосредственно для синтеза нечеткого регулятора.

Ниже предлагается методика лингвистического синтеза нечеткого регулятора с двумя входами и одним выходом для объектов первого и второго порядка при наличии их нечеткого описания.

Пример 1. Лингвистический синтез регулятора для динамического объекта первого порядка. Пусть динамика объекта управления описывается некоторой таблицей лингвистических правил (табл. 2.2). При этом соответствующая лингвистическая переменная, как и ранее, имеет вид: $u^x = \{u^*, W_u, \bar{U}\}$, $u^* \in T^*(u)$, где W_u — универсизм; $T^*(u)$ — расширенное терм-множество типа:

$$\text{если } u^* \text{ есть ОБ и если } x^* \text{ есть НО, тогда } \dot{x}^* \text{ есть ОБ,} \\ \text{иначе, ...} \quad (2.11)$$

Если u^* есть ОБ и если x^* есть ОМ, тогда \dot{x}^* есть ОС, иначе, ...

Если u^* есть ПБ и если x^* есть ПБ, тогда \dot{x}^* есть НО и т.д.

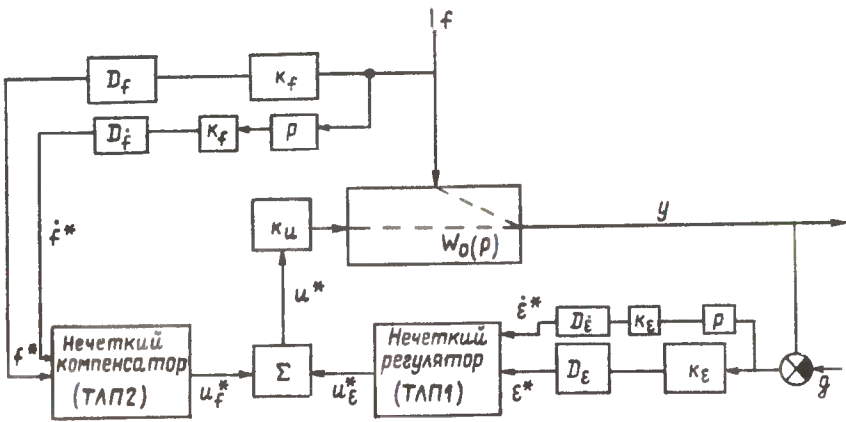


Рис. 4. Структурная схема нечеткой комбинированной системы управления.

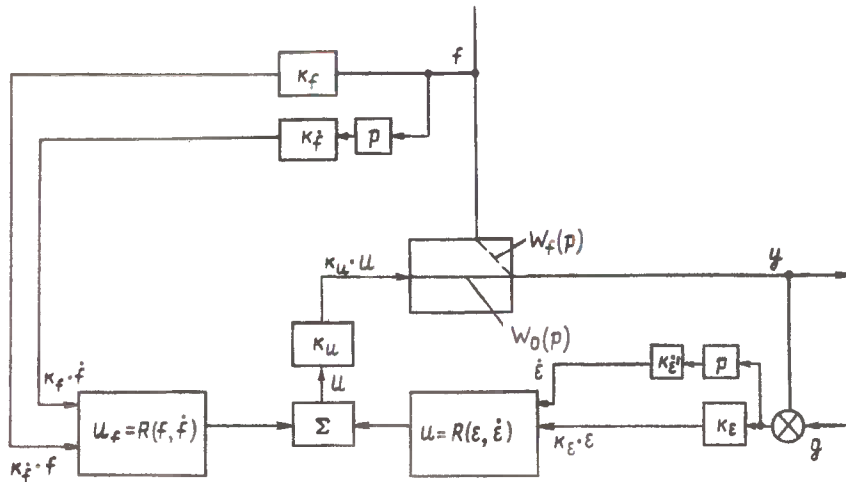


Рис. 5. Структурная схема эквивалентной нечеткой комбинированной системы управления.

Заметим, что вышеуказанный подход может быть распространен и на более сложные многомерные промышленные системы управления.

2.3. Лингвистический синтез динамической системы управления

На практике нередко оказывается трудным получить адекватную модель сложного технологического процесса. В этом случае необходимо проводить определение правил регулирования непосредственно на объекте, что сопряжено не только со

Таблица 2.2

		x^*						
		ОБ	ОС	ОМ	НО	ПМ	ПС	ПБ
u^*	ОБ	НО	ОМ	ОС	ОБ			
	ОС	ПМ	НО	ОМ	ОС			
	ОМ	ПС	ПМ	НО	ОМ			
	НО	ПБ	ПС	ПМ	НО	ОМ	ОС	ОБ
	ПМ				ПМ	НО	ОМ	ОС
	ПС				ПС	ПМ	НО	ОМ
	ПБ				ПБ	ПС	ПМ	НО
			\dot{x}^*					

Таблица 2.3

		g^*						
		ОБ	ОС	ОМ	НО	ПМ	ПС	ПБ
x^*	ОБ	НО	ПМ	ПС	ПБ	ПБ	ПБ	ПБ
	ОС	ОМ	НО	ПМ	ПС	ПБ	ПБ	ПБ
	ОМ	ОС	ОМ	НО	ПМ	ПС	ПБ	ПБ
	НО	ОБ	ОС	ОМ	НО	ПМ	ПС	ПБ
	ПМ	ОБ	ОБ	ОС	ОМ	НО	ПМ	ПС
	ПС	ОБ	ОБ	ОБ	ОС	ОМ	НО	ПМ
	ПБ	ОБ	ОБ	ОБ	ОБ	ОС	ОМ	НО
			e^*					

Таблица лингвистических правил построена на основе реакций объекта на скачкообразные изменения управления на принятом языке. Здесь ОБ, ОС, ОМ, НО, ПМ, ПС, ПБ (отрицательно большое, отрицательно среднее, отрицательно малое, ноль, положительно малое, положительно среднее, положительно большое) — значения (термы) лингвистических переменных.

Таблица 2.6

		e*						
		ОБ	ОС	ОМ	НО	ПМ	ПС	ПБ
u*	ОБ				ПБ	ПС	ПМ	НО
	ОС				ПС	ПМ	НО	ОМ
	ОМ				ПМ	НО	ОМ	ОС
	НО	ПБ	ПС	ПМ	НО	ОМ	ОС	ОБ
	ПМ	ПС	ПМ	НО	ОМ			
	ПС	ПМ	НО	ОМ	ОС			
	ПБ	НО	ОМ	ОС	ОБ			
		e*						

Таблица 2.7

		e*						
		ОБ	ОС	ОМ	НО	ПМ	ПС	ПБ
u*	ОБ				ПБ	ПС	ПМ	НО
	ОС				ПС	ПМ	НО	ОМ
	ОМ				ПМ	НО	ОМ	ОС
	НО	ПБ	ПС	ПМ	НО	ОМ	ОС	ОБ
	ПМ	ПС	ПМ	НО	ОМ			
	ПС	ПМ	НО	ОМ	ОС			
	ПБ	НО	ОМ	ОС	ОБ			
		e*						

Таблица 2.8

		e*						
		ОБ	ОС	ОМ	НО	ПМ	ПС	ПБ
u*	ОБ	ПБ	ПБ		ОБ			НО
	ОС			ОБ	ОС		НО	ПМ
	ОМ		ОБ	ОС	ОМ	НО	ПМ	ПС
	НО	ОБ	ОС	ОМ	НО	ПМ	ПС	ПБ
	ПМ	ОС	ОМ	НО	ПМ	ПС	ПБ	
	ПС	ОМ	НО		ПС	ПБ		
	ПБ	НО			ПБ		ПБ	ПБ
		u*						

Например, система из начального состояния $x_0^n = \text{НО}$; $\dot{x}_0^n = \text{НО}$; $g^n = \text{ПС}$ ($e_0^n = \text{ПС}$; $\dot{e}_0^n = \text{НО}$; $g^n = \text{НО}$) переводится в конечное состояние $x_k^n = \text{ПС}$; $\dot{x}_k^n = \text{НО}$; $g^n = \text{ПС}$ ($\dot{e}_k^n = \text{НО}$; $\dot{e}_k^n = \text{НО}$) с желаемым качеством переходного процесса (табл. 2.4 или табл. 2.5).

Лингвистический синтез нечеткого регулятора можно осуществить по следующей логической схеме [2, 76, 80]

$$(\text{нечеткий объект}) \times (\text{нечеткий регулятор}) = (\text{желаемая замкнутая система}) \quad (2.22)$$

или

$$(\text{нечеткий объект}) \times (\text{нечеткое сравнивающее устройство}) \times (\text{желаемая нечеткая замкнутая система}) = (\text{нечеткий регулятор}). \quad (2.23)$$

Отметим, что схема (2.23) более удобна для лингвистического синтеза, а схема (2.22) — для лингвистического анализа замкнутой системы управления.

Теперь нетрудно вывести лингвистические правила регулирования. Объединив табл. 2.6 с табл. 2.5, получим табл. 2.7, а на основе выражений (2.19), (2.23) и табл. 2.7 имеем:

$$u^* = H^n(e^*, \dot{e}^*), \quad (2.24)$$

т. е. лингвистически синтезированную ТЛП нечеткого регулятора (табл. 2.8).

Таким образом, на основе лингвистического синтеза определена ТЛП (табл. 2.8) нечеткого регулятора, которая имеет вход: ошибку e^* и скорость изменения ошибки \dot{e}^* . Причем неза-

полненную часть табл. 2.8 можно дополнить логически с учетом обратной связи системы.

На основе схемы лингвистического анализа (2.22) покажем, что ТЛП нечеткого регулятора (табл. 2.8) обеспечивает переход системы из начального состояния (2.20) в конечное (2.21): $e_0^n = \text{ПС}$; $\dot{e}_0^n = \text{НО}$; $g^n = \text{ПС}$. При $e_0^n = \text{ПС}$ и $g^n = \text{ПС}$ из табл. 2.3 получаем, что $x^n = \text{НО}$, и из табл. 2.2 следует, что в начальный момент \dot{x}^n также НО, т. е. $\dot{x}^n = \text{НО}$.

При $e^n = \text{ПС}$ и $\dot{e}^n = \text{НО}$ из табл. 2.8 определяем u^n , которое есть ПС. Затем подставляем в табл. 2.2 $u^n = \text{ПС}$ и определяем лингвистические термы x^n и \dot{x}^n . Из табл. 2.2 следует, что возможны 4 состояния: 1) $x^n = \text{НО}$, $\dot{x}^n = \text{ПС}$; или 2) $x^n = \text{ПМ}$, $\dot{x}^n = \text{ПМ}$; или 3) $x^n = \text{ПС}$, $\dot{x}^n = \text{НО}$; или 4) $x^n = \text{ПБ}$, $\dot{x}^n = \text{ОМ}$.

Учитывая, что движение системы будет в сторону увеличения x , можно предположить, что скорость изменения \dot{x} будет неотрицательной. Рассмотрим первые 3 варианта.

1. При $x^n = \text{НО}$, $\dot{x}^n = \text{ПС}$, учитывая уравнения (2.14) и (2.17) определяем, что $e^n = \text{ПС}$ и $\dot{e}^n = \text{ОС}$. При новых значениях e^n и \dot{e}^n из табл. 2.8 имеем $u^n = \text{НО}$. Это означает, что управление остается неизменным, т. е. $u^n = \text{ПС}$.

2. При $x^n = \text{ПМ}$, $\dot{x}^n = \text{ПМ}$ получаем $e^n = \text{ПМ}$, $\dot{e}^n = \text{ОМ}$. При этих значениях e^n и \dot{e}^n из табл. 2.8 определяем, что $u^n = \text{НО}$.

3. При $x^n = \text{ПС}$, $\dot{x}^n = \text{НО}$ получаем $e^n = \text{НО}$, $\dot{e}^n = \text{НО}$. Из табл. 2.8 определяем, что $u^n = \text{НО}$.

Таким образом, нечеткая система переводится из начального состояния (2.20) в конечное (2.21), которое является устойчивым узлом.

Пример 2. Лингвистический синтез основного контура нечеткого регулятора для динамического объекта второго порядка.

Необходимо отметить, что с увеличением порядка объекта его лингвистическое описание (типа табл. 2.2) существенно усложняется. Известно, что объект n -го порядка необходимо представить $(n+2)$ -мерным нечетким отношением в пространстве: $\bar{X}_1 \times \dots \times \bar{X}_2 \times \dots \times \bar{X}_{n+1} \times \bar{U}$, где $\bar{X}_1, \bar{X}_2, \dots, \bar{X}_{n+1}$ и \bar{U} — нечеткие множества соответственно состояний объекта и управления. В частности, лингвистическую модель динамического объекта второго порядка необходимо представить в виде ТЛП с тремя входами, т. е. схематично в виде куба. Очевидно, что составление полной лингвистической модели объекта второго порядка даже с одним входом и выходом — весьма длительная процедура. Чаще всего либо вообще невозможно составить полную $(2n+2)$ -мерную ТЛП, либо при ее составлении могут быть допущены ошибки. Следовательно, синтез лингвистических правил нечеткого регулятора также становится сложным, а при ошибках в ТЛП объекта, естественно, может быть и неверным. Отметим, что если построить нечеткий регулятор с $(2n+2)$ -мерной ТЛП, входами которой будут $x^*, \dot{x}^*, \ddot{x}^*$, а выходом — u^* , то его реализация на микро- и мини-ЭВМ также становится неоправданно сложной.

Для устранения указанных недостатков предлагается способ составления и представления ТЛП динамических объектов второго порядка, позволяющий осуществить лингвистический синтез структуры нечеткого регулятора.

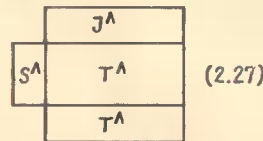
Как известно, переходные процессы линейных и нелинейных устойчивых динамических объектов второго (а также более высокого) порядка характеризуются отрезками *разгон*, *тормоз* и *установившееся состояние*. В связи с этим представляется целесообразным воспользоваться следующим способом описания лингвистических моделей объектов второго порядка с двухходовыми ТЛП. Для определения участка движения (т. е. переходного процесса) объекта введем следующие лингвистические переменные S^n, I^n, T^n , которые определяются через фазовые координаты объекта — x, \dot{x}, \ddot{x} . Причем

$$T^n = R_T^n(S^n, I^n), \quad (2.25)$$

где R_T^n — суть ТЛП. Таким образом, лингвистическая переменная T^n , которая принимает значения из соответствующего термножества.

$$T^*(W_T) = \{\text{разгон}, \text{тормоз}, \text{установившееся состояние}\}, \quad (2.26)$$

идентифицирует участки переходного процесса объекта. Причем она представима в виде следующей ТЛП:



Из анализа уравнений (2.25) — (2.27) следует, что вместо одного нечеткого многомерного отношения $R(x, \dot{x}, \ddot{x}, u)$ удобнее использовать 3 матрицы отношения $R_{\text{разг}}(x, \dot{x}, \ddot{x}, u)$, $R_{\text{торм}}(x, \dot{x}, \ddot{x}, u)$ и $R_{\text{уст}}(x, \dot{x}, \ddot{x}, u)$, размерность которых меньше размерности матрицы отношения $R(x, \dot{x}, \ddot{x}, u)$. Для определения состояния объекта необходимо использовать значения лингвистических переменных T_i^n ($i = \overline{1,3}$), которые определяются в зависимости от значений лингвистических переменных S^n и I^n . Вышесказанное можно представить структурными схемами (рис. 6 и рис. 7).

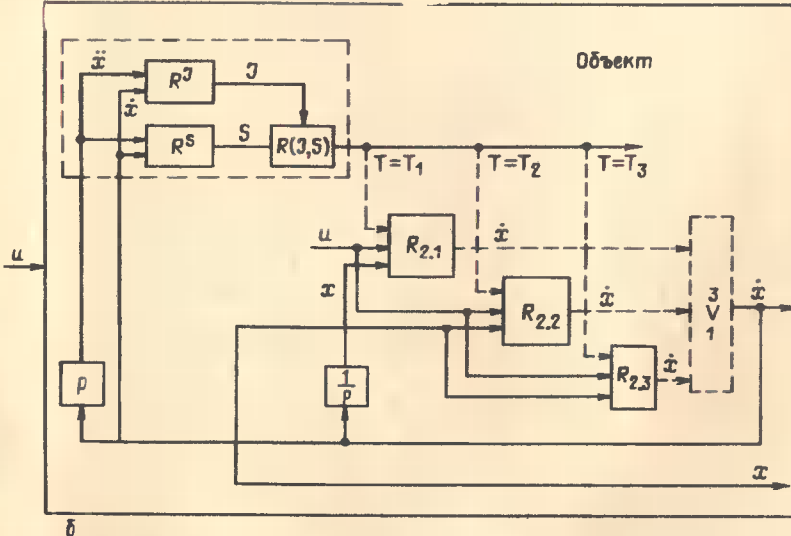
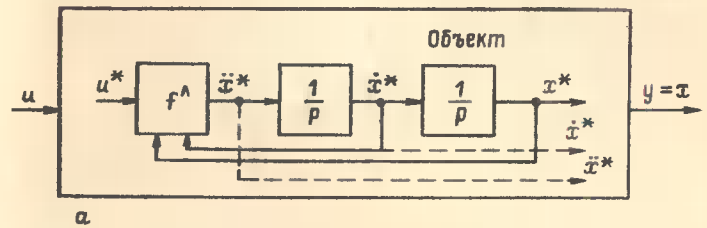


Рис. 6. Структурная схема нечеткой модели динамического объекта управления второго порядка:
а — случай ТЛП с тремя входами; б — случай ТЛП с двумя входами

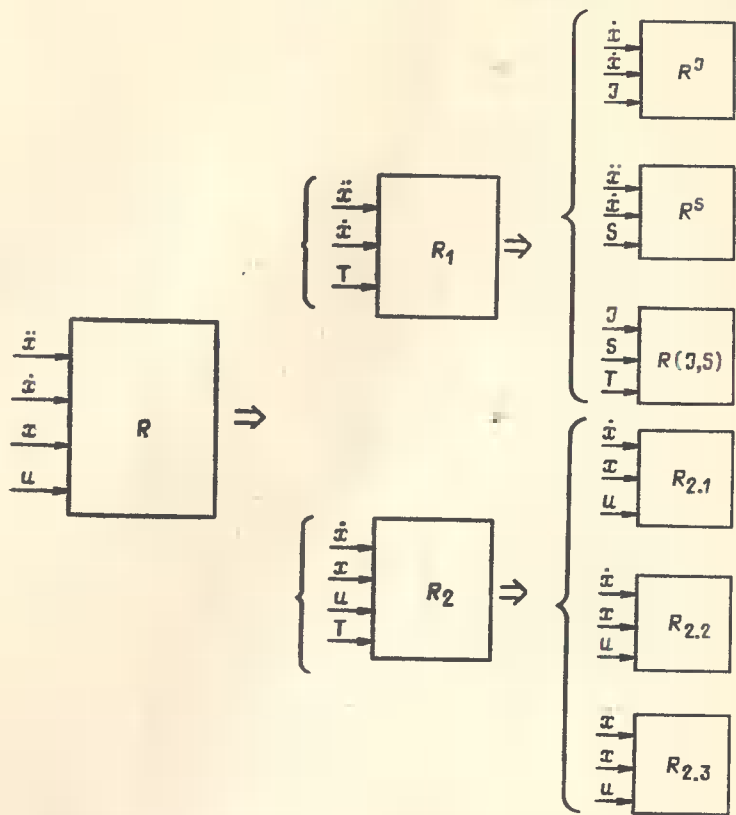


Рис. 7. Пример формирования матриц отношений

Необходимо отметить, что если в области рассуждения каждой переменной выбирается m нечетких подмножеств, то для ТЛП с тремя входами надо составить следующее количество лингвистических правил: $N = m^{n+1}$. При использовании двухвходовых ТЛП количество правил определяется следующим неравенством: $(0,5m^n + 2m) \leq N^1 \leq (2m^n + m)$. Для сравнения трудоемкости составления и запоминания ТЛП в памяти ЭВМ приведем пример. Для ТЛП с тремя входами: при m равном 7 термам, количество правил, N , будет равно 343; при m равном 11 термам, $N = 1341$.

Для декомпозированной ТЛП с двумя входами: при $m = 7$, $39 \leq N^1 \leq 105$; при $m = 11$; $83 \leq N^1 \leq 253$.

Используя лингвистические переменные S^j и I^j , покажем, какими зависимостями от фазовых координат x и \dot{x} они могут

быть представлены. Для объектов второго порядка динамика движения определяется скоростью и ускорением. В связи с этим имеем

$$S^j = \{S^*, W_s, \tilde{S}\} = R_s^j(\dot{x}^*, \ddot{x}^*), S^* \in T(W_s), \quad (2.28)$$

где $S^* = (\text{sign } \dot{x}^*) \wedge (\text{sign } \ddot{x}^*)$; R_s^j — суть ТЛП, \dot{x}^* и \ddot{x}^* соответственно базовые переменные, принимающие значения из набора термов: {положительный, отрицательный, ноль}. Следовательно, согласно выражению (2.28), лингвистическая переменная S^j будет принимать значения из терм-множества:

$$T^*(W_s) = \{\text{ноль, положительный, отрицательный}\}. \quad (2.29)$$

Причем, при положительном S^* переходный процесс находится на участке *разгон*, а при отрицательном S^* — на участке *тормоз*. При нулевом S^* движение объекта может находиться либо в установившемся состоянии, либо в точке перегиба (от значения *тормоз* к значению *разгон* или наоборот). Из последнего следует, что необходима дополнительная информация о соотношении фазовых переменных, на основе которой можно было бы определить установившийся режим. В качестве такой информации рассмотрим значения лингвистической переменной J^{0*} :

$$J^{0*} = \{J^* = R_j^l(\dot{x}^*, \ddot{x}^*), T_j^*, \tilde{J}\}, J^* \in T_s^*;$$

$$J^* = |\overline{\text{sign } \dot{x}^*} \wedge \overline{\text{sign } \ddot{x}^*}|, T_j^* = \{\text{ноль, не ноль}\}, \quad (2.30)$$

где *не ноль* означает, что переходный процесс находится в установившемся состоянии, *ноль* — динамический режим объекта. В таком случае, используя уравнения (2.28) — (2.30) и (2.26), можем предварительно описать участки переходного процесса объекта, которые определим следующими логическими высказываниями:

если S^* есть положительно и если I^* есть ноль, тогда T^* (состояние объекта) — *разгон, иначе...*,

если S^* есть отрицательно и если I^* есть ноль, тогда T^* (состояние объекта) — *тормоз, иначе...*, (2.31)

если S^* есть ноль и если I^* есть не ноль, тогда T^* (состояние объекта) — *установившееся*.

Выражение (2.31) можно представить в виде некоторой ТЛП, входами которой будут лингвистические переменные S^j и I^j , а выходом T^j :

		J^*		
		ноль	НЕ НОЛЬ	
S^*	отрица- тельно	тормоз	(2.32)	
	ноль	вибрационный режим		установившееся
	положи- тельно	разгон		
		T^*		

Отметим, что с помощью (2.32) можно также определять участки переходного процесса динамических объектов более высоких порядков.

Пусть нечеткая модель объекта второго порядка, составленная на основе качественных рассуждений, т. е. по переходным процессам объекта на скачкообразные изменения входов, задана в виде следующих логических высказываний:

если u^* есть ПБ и если x^* есть ПМ, тогда \dot{x}^* есть ПС и \ddot{x}^* есть ПС,
иначе,
если u^* есть ПБ и если x^* есть ПС, тогда \dot{x}^* есть ПБ и \ddot{x}^* есть ПБ,
иначе,

(2.33)

если u^* есть НО и если x^* есть НО, тогда \dot{x}^* есть НО и \ddot{x}^* есть НО.

Допустим также, что переходные процессы (при увеличении отклонения выхода от номинала) объекта второго порядка описываются тремя ТЛП (табл. 2.9, табл. 2.10 и 2.11) соответственно для участков *разгон*, *тормоз* и *установившееся состояние*, а при уменьшении отклонения выхода объекта от номинала — двумя ТЛП (табл. 2.12 и 2.13). Объединив табл. 2.9 и 2.12, а также табл. 2.10 и 2.13, получим обобщенную нечеткую модель соответственно для участков *разгон* и *тормоз* (табл. 2.14 и 2.15). Установившееся состояние объекта в любом случае определяется табл. 2.11 при $T^{II} = T_{22}^{II}$ — *установившееся*.

Сформулируем задачу синтеза нечеткого регулятора для объектов второго порядка (табл. 2.11, 2.14 и 2.15) и с учетом уравнения (2.32) аналогично задаче для объектов 1-го порядка

Таблица 2.9

		x^*						
		ОБ	ОС	ОМ	НО	ПМ	ПС	ПБ
u^*	ОБ	ОС	ОБ	ОС	НО			
	ОС		ОС	ОН	НО			
	ОМ			ОМ	НО			
	НО				НО			
	ПМ				НО	ПМ		
	ПС				НО	ПМ	ПС	
	ПБ				НО	ПС	ПБ	ПС
		\dot{x}^*						
		T^* —разгон						

Таблица 2.10

		x^*						
		ОБ	ОС	ОМ	НО	ПМ	ПС	ПБ
u^*	ОБ	ОМ						
	ОС		ОМ					
	ОМ			ОМ				
	НО				НО			
	ПМ					ПМ		
	ПС						ПМ	
	ПБ							ПМ
		\dot{x}^*						
		T^* —тормоз						

Таблица 2.11

		x^*						
		ОБ	ОС	ОМ	НО	ПМ	ПС	ПБ
u^*	ОБ	НО						
	ОС		НО					
	ОМ			НО				
	НО				НО			
	ПМ					НО		
	ПС						НО	
	ПБ							НО
		\dot{x}^*						
		T^* —установившийся						

Таблица 2.12

		x^*						
		ОБ	ОС	ОМ	НО	ПМ	ПС	ПБ
u^*	ОБ	НО						
	ОС	ПМ	НО					
	ОМ	ПМ	ПС	НО				
	НО	ПМ	ПС	ПБ	НО	ОБ	ОС	ОМ
	ПМ				НО	ОС	ОМ	
	ПС					НО	ОМ	
	ПБ							НО
		\dot{x}^*						
		T^* —разгон						

(формулировку не приводим, так как по форме постановки задач идентичны). Теперь, применяя процедуру синтеза, можем определить структуру и ТЛП нечеткого регулятора для объектов второго порядка. В соответствии с выражениями (2.19), (2.22), (2.23) и с учетом нечеткой модели объекта (см. табл. 2.11, 2.14 и 2.15) получим ТЛП для участков *разгон* и *тормоз* (табл. 2.16 и 2.17). Таким образом, на основании уравнений (2.19), (2.23) и табл. 2.16 и 2.17 синтезируется ТЛП нечеткого регулятора, являющаяся решением поставленной задачи (табл. 2.18, 2.19 и 2.20).

В то же время для простоты реализации нечеткого регулятора целесообразно объединить табл. 2.18 и 2.19 в единую ТЛП,

Таблица 2.13

		x*						
		ОБ	ОС	ОМ	НО	ПМ	ПС	ПБ
u*	ОБ	НО						
	ОС	НО	НО					
	ОМ		ПС	НО				
	НО			ПБ	НО	ОБ		
	ПМ				НО	ОС		
	ПС						НО	НО
	ПБ							НО
		x*						

T* = тормоз

Таблица 2.14

		x*						
		ОБ	ОС	ОМ	НО	ПМ	ПС	ПБ
u*	ОБ	ОС	ОБ	ОС	НО			
	ОС	ПМ	ОС	ОМ	НО			
	ОМ	ПМ	ПС	ОМ	НО			
	НО	ПМ	ПС	ПБ	НО	ОБ	ОС	ОМ
	ПМ				НО	ПМ	ОС	ОМ
	ПС				НО	ПМ	ПС	ОМ
	ПБ				НО	ПС	ПБ	ПС
		x*						

T* = T*_{1,1} = разгон

Таблица 2.15

		x*						
		ОБ	ОС	ОМ	НО	ПМ	ПС	ПБ
u*	ОБ	ОМ						
	ОС	НО	ОМ					
	ОМ		ПБ	ОМ				
	НО			ПБ	НО	ОБ		
	ПМ					ПМ	ОС	
	ПС						ПМ	НО
	ПБ							ПМ
		x*						

T* = T*_{1,3} = тормоз

Таблица 2.16

		e*						
		ОБ	ОС	ОМ	НО	ТМ	ПС	ПБ
u*	ОБ				НО	ПС	ПБ	ПС
	ОС				НО	ПМ	ПС	ОМ
	ОМ				НО	ПМ	ОС	ОМ
	НО	ПМ	ПС	ПБ	НО	ОБ	ОС	ОМ
	ПМ	ПМ	ПС	ОМ	НО			
	ПС	ПМ	ОС	ОМ	НО			
	ПБ	ОС	ОБ	ОС	НО			
		e*						

T* = разгон

Таблица 2.19

		e*						
		ОБ	ОС	ОМ	НО	ПМ	ПС	ПБ
u*	ОБ							
	ОС							
	ОМ							
	НО							
	ПМ							
	ПС							
	ПБ							
		e*						

T* = тормоз

Таблица 2.18

		e*						
		ОБ	ОС	ОМ	НО	ПМ	ПС	ПБ
u*	ОБ							
	ОС							
	ОМ							
	НО							
	ПМ							
	ПС							
	ПБ							
		e*						

T* = разгон

Таблица 2.17

		e*						
		ОБ	ОС	ОМ	НО	ПМ	ПС	ПБ
u*	ОБ							
	ОС							
	ОМ							
	НО							
	ПМ							
	ПС							
	ПБ							
		e*						

T* = разгон

которая представлена табл. 2.21. Следует отметить, что недостающие ситуации в табл. 2.21 дополнялись по смысловому содержанию ТЛП (причем благодаря использованию композиционного правила вывода такое дополнение не обязательно).

Однако при объединении ТЛП могут возникнуть ситуации с наложением резко противоречивых правил. В этом случае предпочтительнее использовать регулятор с тремя ТЛП, который является более «интеллектуальным», но и более сложным в реализации, а лингвистические переменные для выражения (2.32)

Таблица 2.20

		e*						
		OB	OC	OM	HO	PM	PC	PB
e*	OB							
	OC							
	OM							
	HO	PB	PC	PH	HO	OB	OC	OB
	PM							
	PC							
	PB							
			u*					

T* = установившееся

Таблица 2.21

		e*							
		OB	OC	OM	HO	PM	PC	PB	
e*	OB	PB	PB	PC	PM	HO	OM	OC	
	OC	PB	PC	PB	HO	OM	OM	OC	
	OM	PC	PC	PC	OM	OM	OM	OC	
	HO	PC	PC	PM	HO	OM	OC	OC	
	PM	PC	PM	PM	PM	OC	OC	OC	
	PC	PC	PM	PM	PM	OB	OB	OB	
	PB	PC	PM	HO	PM	OC	OB	OB	
			u*						

определяются следующие образом:

$$S_p^n = \{S_p^*, T_s, \tilde{S}_p = \tilde{E} \cap \tilde{E}^i\}, S_p^* \in T_s;$$

$$J_p^n = \{J_p^* = |\text{sign } \dot{e}^*| \wedge |\text{sign } \ddot{e}^*|, T_J, T_p\}, J_p^* \in T_J; \quad (2.34)$$

$$T_p^n = \{T_p^* = R_T(S_p^*, J_p^*), T_p, \tilde{T}_p\}, T_p^* \in T_p.$$

В соответствии с постановкой задачи (2.20)–(2.22) синтезируется ТЛП нечеткого регулятора при известном нечетком описании объекта управления. Назначением синтезированной ТЛП является выработка последовательности u_j^* , $j=0, k-1$, которая с желаемым качеством переходного процесса переводит объект управления из начального состояния (2.20) в конечное (2.21), которое представляет собой устойчивый узел. Эту процедуру можно представить следующие образом:

$$x_0^* \rightarrow u_0^* \rightarrow x_1^* \rightarrow u_1^* \rightarrow \dots \rightarrow x_i^* \rightarrow u_j^* \rightarrow \dots \rightarrow x_k^*, \quad i=0, k, \quad j=0, k-1, \quad (2.35)$$

где x_i^* — выход объекта управления, u_j^* — управляющее воздействие.

Приведенная методика и примеры лингвистического синтеза структуры нечетких регуляторов имеют различные аспекты использования, часть из которых будет изложена ниже.

2.4. Адаптивный нечеткий регулятор

Нечеткие регуляторы, рассмотренные в предыдущих разделах, обеспечивают желаемую переходную характеристику системы управления технологическим процессом лишь при стабильных значениях параметров его математической модели.

Для сохранения заданного качества процессов управления при изменении параметров объекта необходимо корректировать параметры регулятора, в первую очередь, его БЗ. Регуляторы, обладающие указанным свойством приспособления к изменяющейся ситуации с помощью самоорганизации БЗ, будем называть нечеткими адаптивными регуляторами. Структурная схема адаптивного нечеткого регулятора показана на рис. 8. Си-

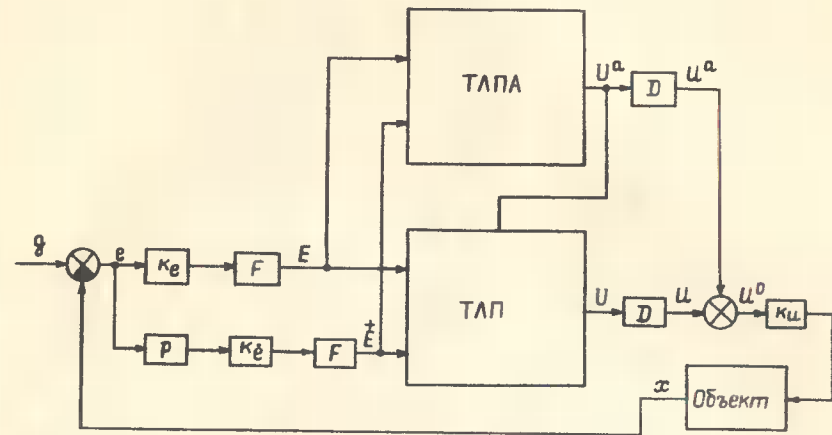


Рис. 8. Структурная схема адаптивного нечеткого регулятора

стема управления с адаптивным регулятором представляет собой двухуровневую систему с обычным нечетким регулятором в цепи ее обратной связи и нечетким контуром адаптации на высшем уровне. Принцип работы адаптивного нечеткого регулятора заключается в том, что при изменении параметров объекта управления в сложившейся текущей ситуации, определяемой ϵ и ϵ^i , из БЗ нечеткого регулятора выбирается действие, не обеспечивающее необходимое качество регулирования.

Получая информацию об изменении параметров объекта, содержащийся в ситуации (ϵ, ϵ^i) верхний контур, изъяв из БЗ правило «уже неудачное», формирует такую последовательность правил, которая приводит к желаемой переходной характеристике системы.

Для решения поставленной задачи организуем контур нечеткой адаптации. Зададим желаемое качество процесса управления в виде подмножества проекций эталонных фазовых траекторий, лежащих в области, в дальнейшем именуемой допустимой. Необходимо отметить, что трудно представить точную фазовую траекторию желаемой реакции объекта управления, а иногда нецелесообразно предъявлять к проектируемой системе такие предельно идеализированные жесткие требования. В этом слу-

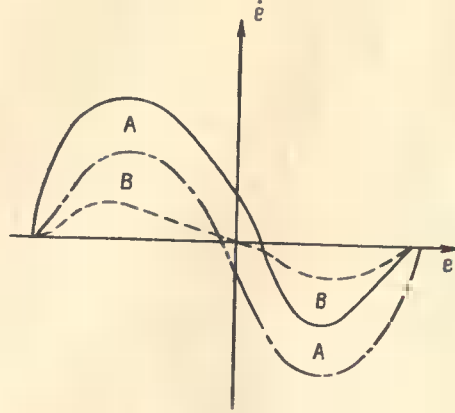


Рис. 9. Вид допустимой области на фазовой плоскости

чае представляется разумным выделить на фазовой плоскости некоторую допустимую область (рис. 9), внутри которой качество работы системы управления воспринималось бы как удовлетворительное и лишь при нарушении границ упомянутой области применялись бы некоторые адаптирующие воздействия. Действительно, очень часто разработчик способен интуитивно выделить допустимую область, либо основываясь на своем опыте, либо из рассуждений о желаемой реакции объекта управления.

Представим эти рассуждения в виде лингвистических правил, например:

если ошибка e^* положительно большая и скорость изменения ошибки \dot{e}^* положительно большая, то переходный процесс системы удовлетворительный и адаптация управления не нужна, иначе ... ,

если ошибка e^* положительно малая и скорость изменения ошибки \dot{e}^* положительно большая, то переходный процесс системы неудовлетворительный и необходимо отрицательно средняя адаптация управления u^* , иначе ... ,

или в виде:

$$R_k^a = \tilde{E}_i \times \tilde{E}_j \times \tilde{U}_l^a, \quad i = \overline{1, n_1}, \quad j = \overline{1, n_2}, \quad l = \overline{1, n_3}, \\ k = \overline{1, n_4}, \quad (2.36)$$

где R_k^a — нечеткое описание правила в пространстве $\tilde{E} \times \tilde{E} \times U^a$, а $\tilde{E}_i, \tilde{E}_j, \tilde{U}_l^a$ — соответственно нечеткие подмножества ошибки, скорости изменения ошибки и адаптации управления. Полученные нечеткие описания правил формируют матрицу нечетких отно-

шений:

$$R^a = R_1^a \vee R_2^a \vee \dots \vee R_k^a \vee \dots \vee R_{n_4}^a, \quad k = \overline{1, n_4}. \quad (2.37)$$

В табл. 2.22 представлена ТЛП контура нечеткой адаптации, с помощью которой производится оценка качества переходного процесса и коррекция правил. Нулевые элементы ТЛП соответствуют состояниям, для которых не требуется коррекция, т. е. переходный процесс удовлетворительный. Эти состояния обеспечивают достаточно быструю установку процесса и успешное затухание в некоторой окрестности задания. Ненулевые элементы ТЛП соответствуют неудовлетворительной реакции объекта управления и необходимости коррекции правила.

Таблица 2.22

		e^*						
		ОБ	ОС	ОМ	НО	ПМ	ПС	ПБ
\dot{e}^*	ОБ	НО	НО	ОС	ОБ	ОБ	ОБ	ОБ
	ОС	НО	НО	НО	ОС	ОС	ОБ	ОБ
	ОМ	НО	ПМ	НО	НО	ОМ	ОС	ОБ
	НО	НО	ПС	ПМ	НО	ОМ	ОС	НО
	ПМ	ПБ	ПС	ПМ	НО	НО	ОМ	НО
	ПС	ПБ	ПБ	ПС	ПС	НО	НО	НО
	ПБ	ПБ	ПБ	ПБ	ПБ	ПС	НО	НО
		u^a						

Таким образом, идея разработчика о желаемой реакции системы представляется единой матрицей R^a , получаемой на основе ТЛП контура нечеткой адаптации. Назначением матрицы R^a является выработка адаптирующего воздействия при нарушении фазовой траекторией системы границы допустимой области. Следует отметить, что предлагаемая ТЛП достаточно универсальна и применима к объектам различного порядка. Однако для нахождения оптимальных (с точки зрения разработчика) границ допустимой области необходимо уделить особое внимание границам двух областей на фазовой плоскости (см. рис. 9): где A — допустимая область; B — область, «ответственная» за время регулирования.

Варьируя при синтезе ТЛП границами областей A и B , разработчик определяет качество переходного процесса, которое должно соответствовать проектируемой САУ.

Как было отмечено, процедура адаптации управления заключается в корректировке некоторого правила из ТЛП основного контура нечеткого регулятора, которое приводит к теку-

щей неудовлетворительной реакции объекта управления. Рассмотрим эту процедуру подробнее.

Допустим, что некоторое правило из ТЛП основного контура нечеткого регулятора явилось причиной текущего неудовлетворительного (с точки зрения ТЛП контура нечеткой адаптации) выхода объекта управления. Нечеткое описание этого правила составляет матрицу:

$$R_{n-k} = \tilde{E}_{n-k} \times \tilde{E}_{n-k} \times \tilde{U}_{n-k}, \quad (2.38)$$

где $\tilde{E}_{n-k} = \tilde{F}(e_{n-k})$, $\tilde{E}_{n-k} = \tilde{F}(\dot{e}_{n-k})$, $\tilde{U}_{n-k} = \tilde{F}(u_{n-k})$, n — индекс текущего времени опроса объекта, k определяется из динамики, а F — оператор перехода от четкой переменной к нечеткой. Текущая неудовлетворительная реакция объекта управления была установлена некоторым правилом из ТЛП контура адаптации, нечеткое описание которого также составляет матрицу:

$$R_n^a = \tilde{E}_n \times \tilde{E}_n \times \tilde{U}_n^a. \quad (2.39)$$

Очевидно, что неудовлетворительное правило (2.38) должно быть заменено скорректированным правилом:

$$R' = \tilde{E}_{n-k} \times \tilde{E}_{n-k} \times \tilde{U}^H, \quad (2.40)$$

где $\tilde{U}^H = \tilde{F}(u_{n-k} + u_n^a)$.

Таким образом, в ТЛП основного контура будет занесено новое правило, которое для текущих параметров объекта управления предпочтительнее удаленного правила. А для решения вопроса адаптации текущего неудовлетворительного управления представляется целесообразным использовать четкое значение u в соответствии со следующим алгоритмом: $u^0 = u_n + u_n^a$, где $u_n = \mathcal{D}\{U_n\}$, $u_n^a = \mathcal{D}\{U_n^a\}$, а \mathcal{D} — оператор перехода от нечеткой переменной к четкой.

С учетом изложенного и формировалась структурная схема нечеткого регулятора с контуром нечеткой адаптации, представленная на рис. 8.

Необходимо отметить возможность применения контура нечеткой адаптации для первоначальной генерации управляющих правил ТЛП основного контура при условии стационарности параметров объекта управления [85, 86]. В этом случае используется программная имитация системы управления и до начала эксперимента ТЛП основного контура не содержит правил. Эксперимент заключается в том, что регулятору устанавливается задание и на некотором интервале времени подсчитывается число сгенерированных правил. Далее эксперимент повторяется до полной выработки алгоритма и прекращения коррекции правил. После окончания эксперимента ТЛП основного контура содержит необходимое количество правил для вывода объекта управления на установленное задание с желаемым качеством переходного процесса.

В заключение отметим, что благодаря использованию алгоритма быстрого вывода [87] суммарное время выполнения процедуры нечеткой адаптации к выработке управляющего воздействия не превышает время выработки управляющего воздействия в нечетком регуляторе, использующем обычное композиционное правило вывода.

В следующем разделе кратко рассмотрим применение изложенных методов к построению промышленных нечетких САУ.

2.5. Примеры внедренных нечетких промышленных динамических систем управления

2.5.1. Нечеткая адаптивная система управления ректификационной колонной

Как уже отмечалось, процедура синтеза нечеткого регулятора заключается в определении множества лингвистических правил управления и значений коэффициентов масштабирования k_e , k_e и k_u , обеспечивающих желаемое качество процесса управления. Для нечеткого адаптивного регулятора необходимо дополнительно определить множество лингвистических правил адаптации управления. Поскольку ТЛП основного контура и контура нечеткой адаптации имеют идентичные входы, то в нечетком адаптивном регуляторе рассматриваются 4 универсальных множества, на которых соответственно определены следующие нечеткие подмножества:

$$\tilde{E}_i (i = \overline{1,14}), \tilde{E}_j (j = \overline{1,13}), \tilde{U}_l (l = \overline{1,13}), \tilde{U}_k^a (k = \overline{1,13}),$$

где \tilde{E}_i , \tilde{E}_j , \tilde{U}_l и \tilde{U}_k^a — соответственно нечеткие подмножества ошибки, управления и адаптации управления. Указанные нечеткие подмножества определены для соответствующих лингвистических термов: $u_1^* = \text{ОБ (отрицательно большое)} \triangleq (u, \mu_1(u))$; $u_2^* = \text{ОНБ (отрицательно ниже большого)} \triangleq (u, \mu_2(u))$; $u_3^* = \text{ОС (отрицательно среднее)} \triangleq (u, \mu_3(u))$; $u_4^* = \text{ОНС (отрицательно ниже среднего)} \triangleq (u, \mu_4(u))$; $u_5^* = \text{ОМ (отрицательно малое)} \triangleq (u, \mu_5(u))$; $u_6^* = \text{ОММ (отрицательно ниже малого)} \triangleq (u, \mu_6(u))$; $u_7^* = \text{НО (ноль)} \triangleq (u, \mu_7(u))$; $u_8^* = \text{ПМ (положительно ниже малого)} \triangleq (u, \mu_8(u))$; $u_9^* = \text{ПМ (положительно малое)} \triangleq (u, \mu_9(u))$; $u_{10}^* = \text{ПМС (положительно ниже среднего)} \triangleq (u, \mu_{10}(u))$; $u_{11}^* = \text{ПС (положительно среднее)} \triangleq (u, \mu_{11}(u))$; $u_{12}^* = \text{ПНБ (положительно ниже большого)} \triangleq (u, \mu_{12}(u))$; $u_{13}^* = \text{ПБ (положительно большое)} \triangleq (u, \mu_{13}(u))$.

Здесь u_i^* ($i = \overline{1, 13}$) — лингвистический терм соответствующей лингвистической переменной нечеткого подмножества \tilde{U}_l , $\mu_l(u)$ — функция принадлежности нечеткого подмножества \tilde{U}_l .

Аналогичным образом определены на принятом языке нечеткие подмножества для скорости изменения ошибки: $\tilde{E}_j \triangleq (\dot{e}_M, \mu_j(\dot{e}_M))$, $j = \overline{1, 13}$; адаптация управления: $U_k^a \triangleq (u^a, \mu_k(u^a))$, $k = \overline{1, 13}$; и ошибки управления: $E_i \triangleq (e_M, \mu_i(e_M))$, $i = \overline{1, 14}$. Количество лингвистических термов ошибки на единицу больше количества других термов вследствие того, что вместо термина НО для ошибки используются 2 термина ПН и ОН (положительный ноль и отрицательный ноль).

В приведенных выражениях масштабированные величины e_M и \dot{e}_M определены следующим образом: $e_M = k_e \cdot e$; $\dot{e}_M = k_{\dot{e}} \cdot \dot{e}$. На рис. 10 представлены универсальные множества (общие пределы изменения) и графики функций принадлежности $\mu_l(e_M)$, $\mu_j(\dot{e}_M)$, $\mu_i(u)$ и $\mu_k(u^a)$, соответствующие ошибке E_i ($i = \overline{1, 14}$), скорости изменения ошибки \tilde{E}_j ($j = \overline{1, 13}$) управлению \tilde{U}_l ($l = \overline{1, 13}$) и адаптации управления \tilde{U}_k^a ($k = \overline{1, 13}$). Для всех функций принадлежности выбрана единая экспоненциальная форма:

$$\begin{aligned} \mu_l(e_M) &= \exp\{-q_{1l}|e_M - \bar{e}_{Ml}|\}; & \mu_j(\dot{e}_M) &= \exp(-q_{2j}|\dot{e}_M - \bar{\dot{e}}_{Mj}|), \\ \mu_i(u) &= \exp\{-q_{3i}|u - \bar{u}_i|\}; & \mu_k(u^a) &= \exp\{-q_{4k}|u^a - \bar{u}_k^a|\}; \end{aligned} \quad (2.41)$$

$$i = \overline{1, 14}, \quad j = \overline{1, 13}, \quad l = \overline{1, 13}, \quad k = \overline{1, 13},$$

где \bar{e}_{Ml} , $\bar{\dot{e}}_{Mj}$, \bar{u}_i и \bar{u}_k^a соответствуют средним точкам i -го, j -го l -го и k -го подмножества ошибки, скорости изменения ошибки, управления и адаптации управления, степень принадлежности которых равна 1; а q_{1l} , q_{2j} , q_{3i} и q_{4k} — константы, определяемые из выполнения следующих условий:

$$\begin{aligned} \mu_l(e_M) &\geq 0,5, \quad e_M \in E_l, \quad l = \overline{1, 14}; & \mu_j(\dot{e}_M) &\geq 0,5, \quad \dot{e}_M \in \tilde{E}_j, \quad j = \overline{1, 13}; \\ \mu_i(u) &\geq 0,5, \quad u \in U_l, \quad l = \overline{1, 13}; & \mu_k(u^a) &\geq 0,5, \quad u^a \in U_k^a, \quad k = \overline{1, 13}. \end{aligned} \quad (2.42)$$

В табл. 2.23 приведены значения q_{1l} , q_{2j} , q_{3i} , q_{4k} , \bar{e}_{Ml} , $\bar{\dot{e}}_{Mj}$, \bar{u}_i и \bar{u}_k^a .

С помощью предложенных выше подходов были синтезированы ТЛП основного контура регулятора и ТЛП контура нечеткой адаптации. В табл. 2.24 рассмотрена ТЛП основного контура, а в табл. 2.25 — ТЛП контура нечеткой адаптации.

Блок-схема программной реализации разработанного нечеткого адаптивного регулятора для управления температурой верхней части колонны К-2 приведена на рис. 11. Алгоритм функционирует следующим образом.

БЛОК 1. Регулятору устанавливается задание g_i .

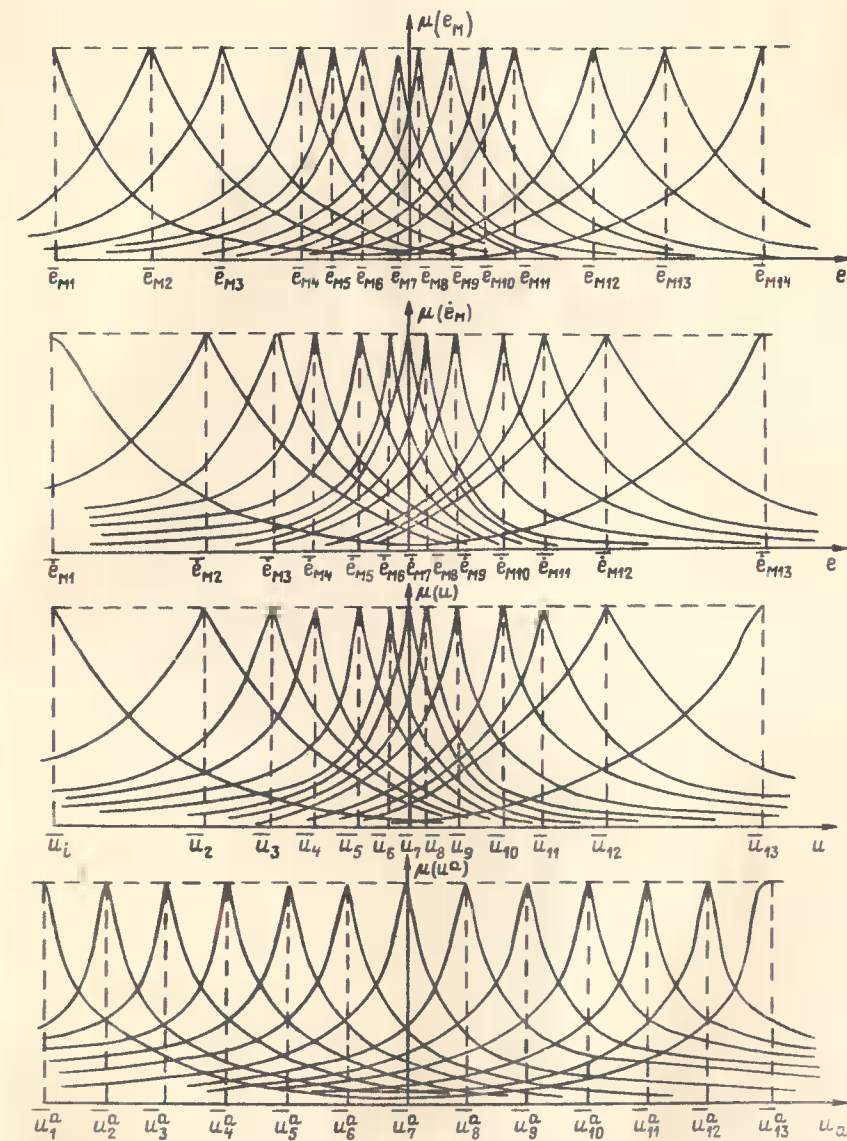


Рис. 10. Графики функций принадлежности

Таблица 2.23

i	q_{1i}	q_{2j}	q_{3l}	q_{4k}	\bar{e}_{Mi}	\bar{e}_{Mj}	\bar{u}_l	\bar{u}_k^a
1	2,0117	0,4952	4,0235	107,2933	-10	-20	-0,3	-0,180
2	2,6823	0,6705	16,094	107,2933	8,6	-11,1	-0,25	-0,165
3	2,1458	1,4630	100,5875	107,2933	-7,25	-7,6	-0,225	-0,135
4	1,2875	1,3994	128,752	107,2933	-5,25	-5,35	-0,2	-0,105
5	1,6094	1,2380	107,2933	107,2933	3	-2,9	-0,15	-0,075
6	2,1458	2,1458	128,752	107,2933	-1,25	-0,85	-0,125	-0,045
7	6,4376	16,094	32,188	107,2933	-0,25	0	0	0
8	6,4376	2,1458	128,752	107,2933	0,25	0,85	0,125	0,045
9	2,1458	1,2380	107,2933	107,2933	1,25	2,9	0,15	0,075
10	1,6094	1,3994	128,752	107,2933	3	5,35	0,2	0,105
11	1,2875	1,4630	100,5875	107,1933	5,25	7,6	0,225	0,135
12	2,1458	0,6705	16,094	107,2933	7,25	11,1	0,25	0,165
13	2,6823	0,4952	4,0235	107,2933	8,6	20	0,3	0,180
14	2,0117	—	—	—	10	—	—	—

Таблица 2.24

		e^*													
		ОБ	ОНБ	ОС	ОНС	ОМ	ОНМ	ОН	ПН	ПНМ	ПМ	ПНС	ПС	ПНБ	ПБ
e	ОБ	ПБ	ПНБ	ПНБ	ПНБ	ПНБ	ПНБ	ПНБ	ОНБ	ОНБ	ОНБ	ОНБ	ОНБ	ОБ	ОБ
	ОНБ	ПБ	ПНБ	ПНБ	ПНБ	ПНБ	ПНБ	ПНБ	ОНБ	ОНБ	ОНБ	ОНБ	ОНБ	ОБ	ОБ
	ОС	ПБ	ПНБ	ПНБ	ПНБ	ПНБ	ПНБ	ПС	ОС	ОС	ОС	ОС	ОНБ	ОНБ	ОБ
	ОНС	ПБ	ПНБ	ПНБ	ПНБ	ПС	ПНС	ПНС	ПНМ	ОНС	ОНС	ОНС	ОС	ОНБ	ОБ
	ОМ	ПБ	ПНБ	ПНБ	ПС	ПНС	ПМ	ПНМ	ПМ	ОНМ	ОМ	ОНС	ОС	ОНБ	ОБ
	ОНМ	ПБ	ПНБ	ПС	ПНС	ПМ	ПНС	ПМ	НО	НО	НО	ОМ	ОНС	ОС	ОБ
	НО	ПБ	ПНБ	ПС	ПНС	ПМ	ПМ	ПМ	НО	НО	ОМ	ОНС	ОС	ОНБ	ОБ
	ПНМ	ПБ	ПНБ	ПС	ПНС	ПС	ПНС	ПНМ	НО	НО	ОНМ	ОНС	ОС	ОНБ	ОБ
	ПМ	ПБ	ПНБ	ПНС	ПНС	ПНС	ПНС	ПНС	ОНС	ОНС	ОС	ОНБ	ОБ	ОБ	ОБ
	ПНС	ПБ	ПНБ	ПС	ПС	ПС	ПС	ПС	ОС	ОС	ОНБ	ОБ	ОБ	ОБ	ОБ
	ПС	ПБ	ПНБ	ПНБ	ПНБ	ПНБ	ПНБ	ПНБ	ОНБ	ОНБ	ОС	ОБ	ОБ	ОБ	ОБ
	ПНБ	ПБ	ПБ	ПБ	ПБ	ПБ	ПБ	ПНБ	ОНБ	ОБ	ОБ	ОБ	ОБ	ОБ	ОБ
	ПБ	ПБ	ПБ	ПП	ПБ	ПБ	ПБ	ПБ	ОБ	ОБ	ОБ	ОБ	ОБ	ОБ	ОБ

 u^*

Таблица 2.25

		e^*															
		ОБ	ОНБ	ОС	ОНС	ОМ	ОНМ	ОН	ПН	ПНМ	ПМ	ПНС	ПС	ПНБ	ПБ		
e	ОБ	НО	НО	НО	НО	НО	НО	НО	ПНМ	ПМ	ПНС	ПС	ПНБ	ПБ	ПБ		
	ОНБ	НО	НО	НО	НО	НО	НО	НО	ПНМ	ПНС	ПС	ПНБ	ПНБ	ПБ	ПБ		
	ОС	НО	НО	НО	НО	НО	НО	НО	ПНМ	ПМ	ПНС	ПС	ПНБ	ПБ	ПБ		
	ОНС	НО	НО	ОМ	НО	НО	НО	НО	ПНМ	ПНМ	ПМ	ПНС	ПС	ПБ	ПБ		
	ОМ	НО	ОМ	ОС	ОМ	НО	НО	НО	НО	ПНМ	ПМ	ПС	ПБ	ПБ	ПБ		
	ОНМ	НО	ОНС	ОНБ	ОНС	НО	НО	НО	НО	ПНМ	ПМ	ПС	ПБ	ПБ	ПБ		
	e^*	НО	НО	ОБ	ОБ	ОС	ОМ	ОНМ	НО	НО	ПНМ	ПМ	ПС	ПБ	ПБ	НО	
	ПНМ	ОБ	ОБ	ОБ	ОС	ОМ	ОНМ	НО	НО	НО	НО	ПНС	ПНБ	ПНС	НО		
	ПМ	ОБ	ОБ	ОБ	ОС	ОМ	ОНМ	НО	НО	НО	НО	ПМ	ПС	ПМ	НО		
	ПНС	ОБ	ОБ	ОБ	ОС	ОНС	ОН	ОНМ	ОНМ	НО	НО	НО	НО	НО	НО		
	ПС	ОБ	ОБ	ОБ	ОНБ	ОС	ОНС	ОМ	ОНМ	НО	НО	НО	НО	НО	НО		
	ПНБ	ОБ	ОБ	ОБ	ОНБ	ОНБ	ОС	ОНС	ОНМ	НО	НО	НО	НО	НО	НО		
	ПБ	ОБ	ОБ	ОБ	ОБ	ОБ	ОНБ	ОС	ОНС	ОМ	ОНМ	НО	НО	НО	НО		

 u^*

БЛОК 2. На основе вновь полученного задания g_i и предыдущего задания рассчитываются коэффициенты масштабирования k_e , k_z и k_u .

БЛОК 3. В результате опроса объекта управления в ЭВМ вводится значение температуры верхней части колонны $K-2$.

БЛОК 4. Вычисляются значения e и e^* , а в результате их масштабирования — значения e_M и e_{M^*} .

БЛОК 5. Вычисляются функции принадлежности $\mu(e_M)$ и $\mu(e_{M^*})$ соответствующих нечетких подмножеств ошибки и скорости изменения ошибки.

БЛОК 6. Дается нечеткая оценка качества переходного процесса. В случае выявления удовлетворительного качества переходного процесса осуществляется переход на БЛОК 8.

БЛОК 7. В случае выявления неудовлетворительного качества переходного процесса корректируется соответствующее правило и в ТЛП основного контура заносится новое правило.

БЛОК 8. С использованием быстрого алгоритма вывода вычисляется функция принадлежности нечеткого подмножества управления $\mu(u)$.

БЛОК 9. На основе полученной функции принадлежности $\mu(u)$ выбирается значение управления u .

БЛОК 10. Управление u выдается на объект управления.

Процедура конструирования нечеткого адаптивного регулятора была проведена для канала регулирования «расход острого орошения — температура верхней части колонны К-2». Аналогичная процедура конструирования была проведена и для других каналов регулирования.

Однако наличие перекрестных связей в объекте управления дополнительно требует выполнения условия автономности. В связи с этим проведен расчет компенсаторов перекрестных связей. Была выбрана относительно простая структура компенсаторов, которая определялась следующим образом [88]:

$$W_{ij}^A(z) = -\frac{W_{ij}(z)}{W_{ii}(z)}, \quad i = \overline{1, 4}, \quad j = \overline{1, 4}, \quad i \neq j. \quad (2.43)$$

Естественно, что из-за присущей объекту управления нестационарности такой тип компенсатора не может обеспечить полную автономность. Однако из-за простоты структуры и возможности практической реализации на мини-ЭВМ этот компенсатор является наиболее предпочтительным. Как показал анализ функционирования системы управления, компенсация взаимовлияния каналов регулирования была вполне приемлемой. Таким образом, подставляя в выражение (2.43) значения передаточных функций (предварительно перейдя к дискретному виду), можно определить передаточные функции компенсатора, обеспечивающие условие автономности.

Структурная схема разработанной нечеткой адаптивной САУ атмосферным блоком установки ЭЛОУ АВТ приведена на рис. 12.

Учитывая необходимость апробации разработанной САУ, а также необходимость проведения сравнительного анализа с существующими системами, было проведено всестороннее исследование нечеткой адаптивной системы методами программной имитации. Для моделирования использовался алгоритмический язык ФОРТРАН и все эксперименты были выполнены на управляющем вычислительном комплексе СМ-4 под управлением операционной системы ФОБОС.

Ниже показаны наиболее характерные результаты проведенных экспериментов с нечетким адаптивным регулятором, нечетким регулятором и регуляторами, реализующими традиционные законы регулирования.

Следуя схеме синтеза нечеткой адаптивной системы первоначально были получены управляющие правила основного контура регулятора. В номинальном режиме установок динамика каналов регулирования приближенно аппроксимируется дифференциальным уравнением второго порядка. С помощью подхода, приведенного в подразделе 2.3, были получены 3 матрицы нечеткого регулятора: матрица для участка тормоз (Т-матрица), для участка разгон (Р-матрица) и для участка установившееся состояние (У-матрица). Определены универ-

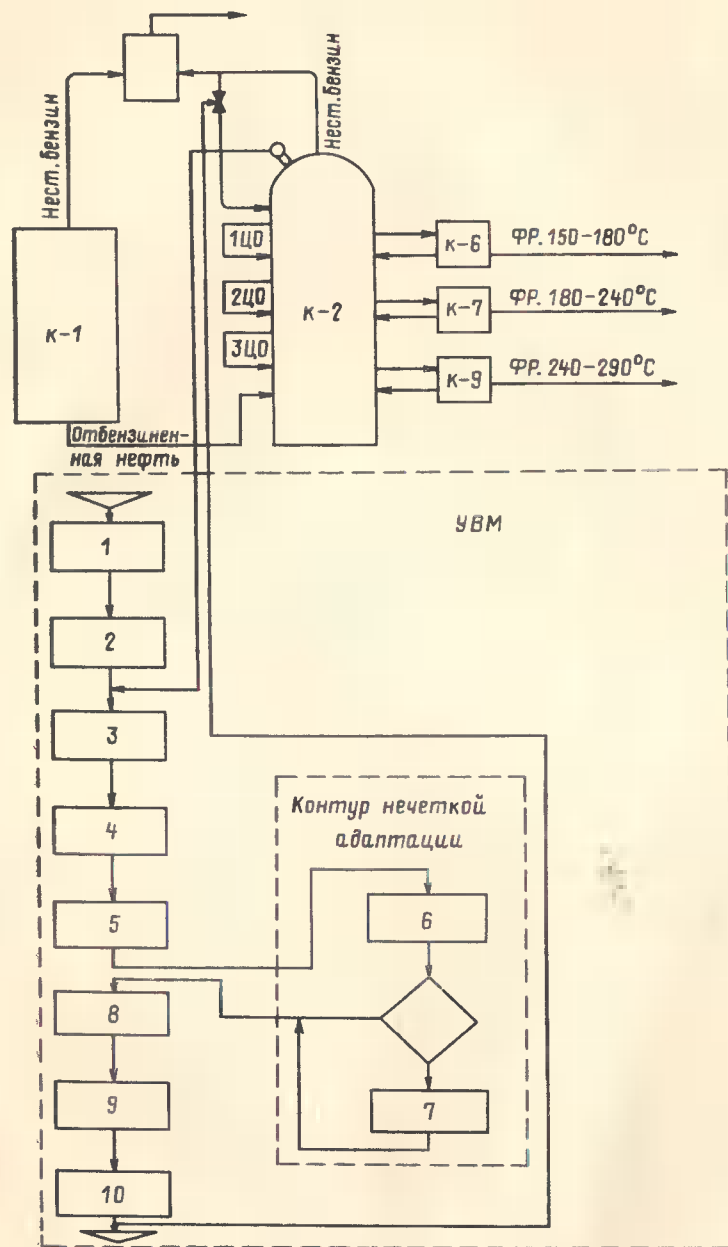


Рис. 11. Блок-схема нечеткого адаптивного регулятора

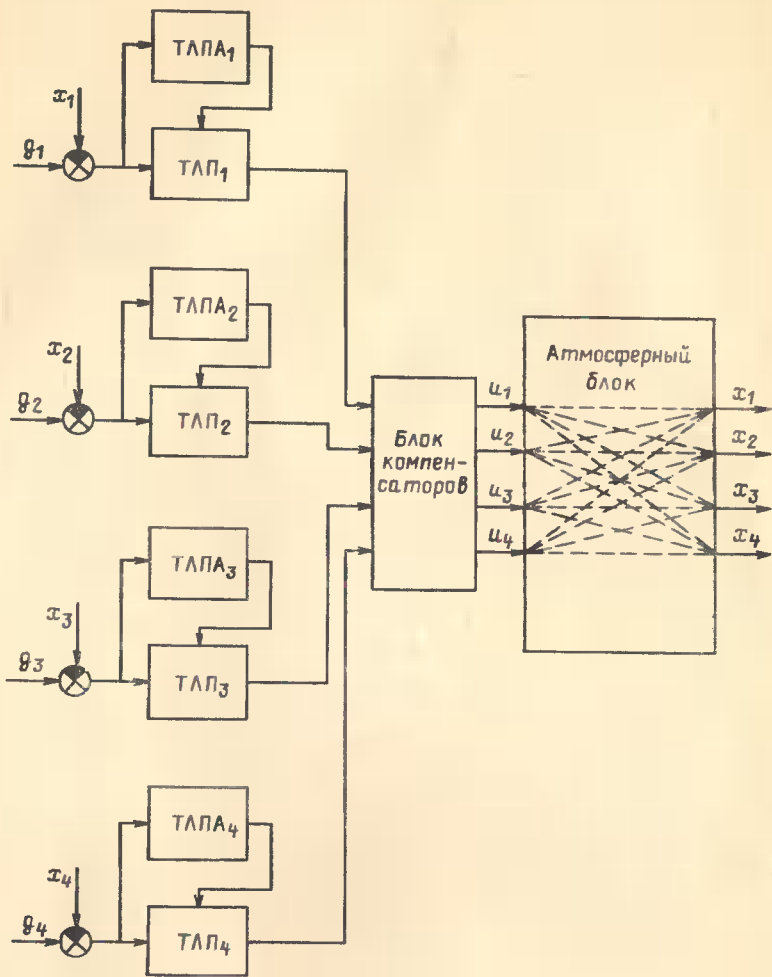


Рис. 12. Структурная схема нечеткой адаптивной САУ

сальные множества ошибки, скорости изменения ошибки и управления. Интервал дискретности определен, как и для реального объекта, и равнялся 15 с.

Процедура эксперимента заключалась в том, что с учетом значений и знака фазовых координат использовалось управляющее правило, входящее в одну из трех имеющихся матриц.

На рис. 13 приведена блок-схема реализации данного регулятора. Работа блоков осуществляется следующим образом.

БЛОК 1. Регулятору устанавливается задание g_i .

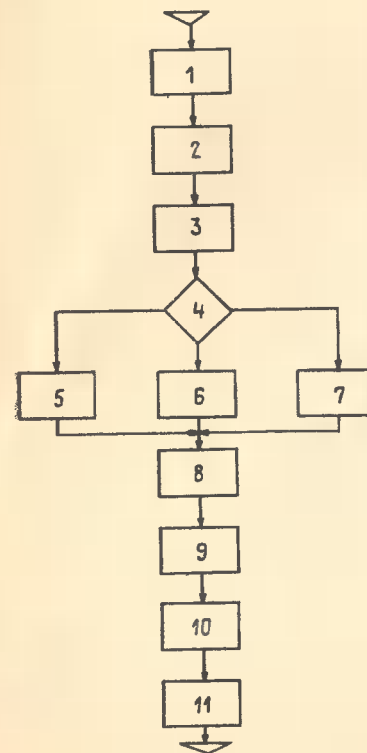


Рис. 13. Блок-схема реализации работы регулятора

БЛОК 2. На основе вновь полученного задания g_i и предыдущего задания рассчитываются коэффициенты масштабирования k_e , $k_{\dot{e}}$ и $k_{\ddot{e}}$.

БЛОК 3. Вычисляются значения e , \dot{e} и \ddot{e} , а в результате масштабирования — значения e_M и \dot{e}_M .

БЛОК 4. Анализируются значения \dot{e} и \ddot{e} .

БЛОК 5. Для управления активизируется Т-матрица.

БЛОК 6. Для управления активизируется Р-матрица.

БЛОК 7. Для управления активизируется У-матрица.

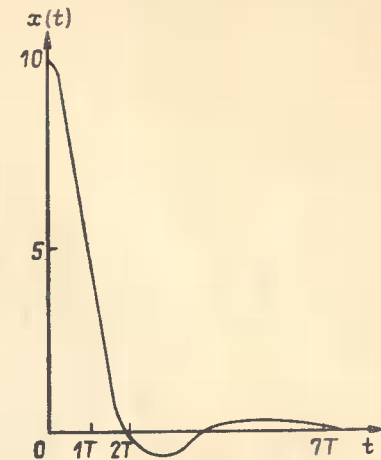


Рис. 14. Вид переходного процесса регулятора

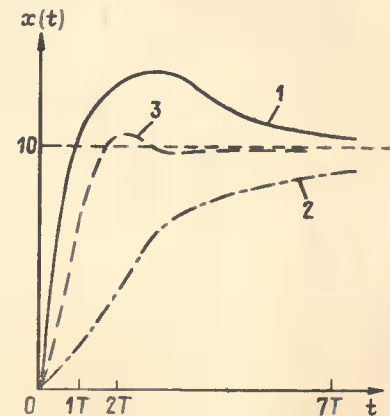


Рис. 15. Вид переходного процесса регуляторов

БЛОК 8. Вычисляются функции принадлежности $\mu(e_M)$ и $\mu(\dot{e}_M)$ соответствующих нечетких подмножеств ошибки и скорости изменения ошибки.

БЛОК 9. Вычисляется функция принадлежности нечеткого подмножества управления $\mu(u)$.

БЛОК 10. На основе подмножества $\mu(u)$ выбирается значение управления u .

БЛОК 11. Значение u выдается на управление.

Эксперименты, проведенные с этим регулятором, выявили хорошее качество переходного процесса (рис. 14). Однако существенным недостатком такой структуры нечеткого регулятора являются значительные затраты оперативной памяти ЭВМ, что обусловлено необходимостью хранения трех матриц.

В связи с этим была выполнена процедура объединения трех ТЛП, причем в случае наложения правил выбирался компромиссный вариант. Использование единой ТЛП практически не ухудшило качество процесса управления, и нечеткий регулятор продемонстрировал малую чувствительность к изменению параметров модели объекта в определенном диапазоне. Так, например, приведенный сравнительный анализ между стандартными регуляторами и полученным нечетким регулятором выявил преимущество последнего при $\pm 10\%$ -ной вариации коэффициента усиления объекта. На рис. 15 представлены кривые переходного процесса, полученные ПИ-регулятором (кривые 1 и 2) и нечетким регулятором (кривая 3). Блок-схема реализации нечеткого регулятора с единой ТЛП представлена на рис. 16. Работа блоков осуществляется следующим образом.

БЛОК 1. Регулятору устанавливается задание.

БЛОК 2. Рассчитываются коэффициенты k_e , $k_{\dot{e}}$ и k_u .

БЛОК 3. Вычисляются значения e_M и \dot{e}_M .

БЛОК 4. Вычисляются функции принадлежности $\mu(e_M)$ и $\mu(\dot{e}_M)$.

БЛОК 5. Вычисляется функция принадлежности $\mu(u)$.

БЛОК 6. На основе функции принадлежности $\mu(u)$ выбирается значение управления u .

БЛОК 7. Значение u выдается на управление.

Как уже было отмечено, некоторая вариация коэффициента усиления объекта (до $\pm 10\%$) незначительно влияла на качество переходного процесса. Однако исследования реального объекта позволяют предположить возможность более значительной (до $\pm 50\%$) вариации параметров атмосферного блока. Этот факт обусловил необходимость исследования характеристик разработанного нечеткого регулятора в широком диапазоне изменения параметров модели объекта управления.

Как показали результаты экспериментов, наличие существенных параметрических возмущений значительно ухудшило качество процесса управления. На рис. 17 приведены кривые

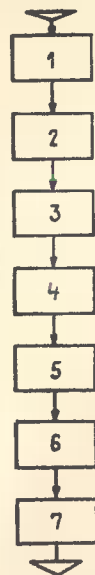


Рис. 16

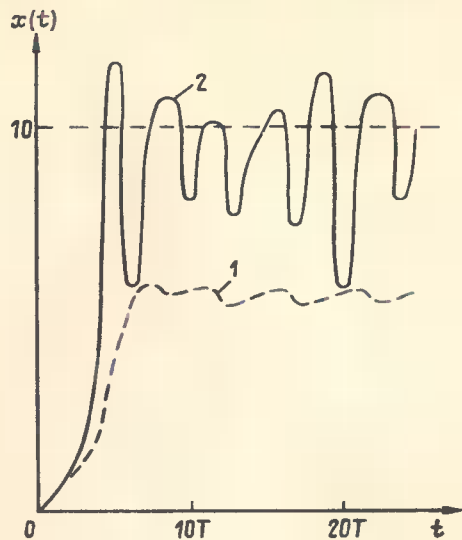


Рис. 17

Рис. 16. Блок-схема реализации работы нечеткого регулятора с единой ТЛП

Рис. 17. Кривые переходного процесса регулятора

переходного процесса, зафиксированные при $k_{об}=30$ (кривая 1) и $k_{об}=80$ (кривая 2).

Таким образом, результаты имитационного моделирования на ЭВМ выявили ограниченную применимость нечеткого регулятора для данного объекта. В связи с этим нечеткий регулятор был дополнен контуром нечеткой адаптации и была проведена новая серия экспериментов на ЭВМ при аналогичной вариации параметров объектов управления. В результате отмечено удовлетворительное качество переходного процесса даже при предельных значениях параметров объекта. На рис. 18 приведены сравнительные характеристики нечеткого регулятора (кривая 1) и нечеткого адаптивного регулятора (кривая 2) при значении коэффициента усиления объекта $k_{об}=80$.

Контур нечеткой адаптации, как отмечалось, можно использовать и для первоначальной генерации управляющих правил основного контура регулятора. В этом случае суть эксперимента заключалась в том, что проводился сравнительный анализ качества процесса управления, который обеспечивали для идентичной модели объекта 2 нечетких регулятора. ТЛП одного из них была синтезирована эвристически, исходя из желаемой реакции системы управления. ТЛП другого регулятора синтезировалась эвристическим путем, исходя из желаемой реакции системы управления, которая задавалась контуром нечеткой адаптации.

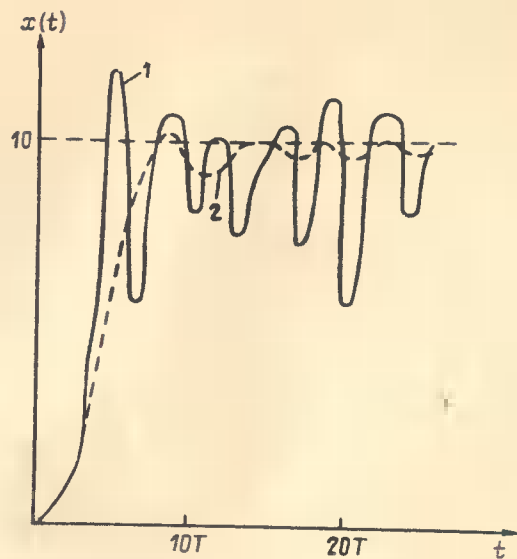


Рис. 18. Характеристики качества управления:
1 — нечеткий регулятор; 2 — нечеткий адаптивный регулятор

На рис. 19 приведены результаты переходного процесса для эвристически синтезированной ТЛП (кривая 1) и для автоматически синтезируемой ТЛП (кривая 2), показывающие преимущество контура нечеткой адаптации. Однако такой подход

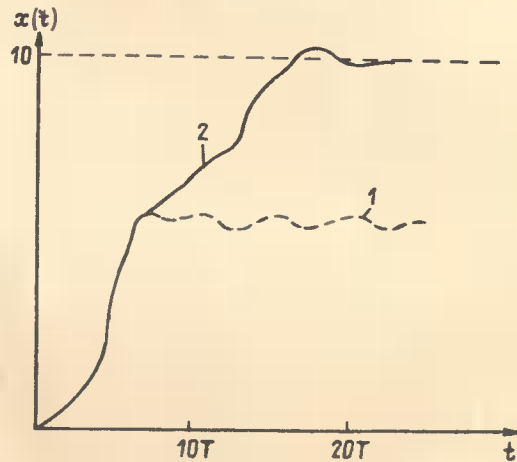


Рис. 19. Результаты моделирования переходных процессов регуляторов:
1 — регулятор с эвристически синтезированной ТЛП; 2 — регулятор с автоматически синтезированной ТЛП

к синтезу ТЛП возможен лишь при наличии адекватного аналитического описания объекта управления методами программной имитации.

Рассмотренные примеры относились к классу объектов, традиционных для классической теории автоматического регулирования и управления.

Обсудим теперь применение изложенных методов анализа нечетких моделей к классу систем, которым присущ процесс принятия решений.

2.5.2. Нечеткие модели принятия решений в интеллектуальных роботах

Разработчики систем управления в настоящее время все чаще обращаются к идее создания гибких автоматизированных комплексов и целых производств, так называемых гибких производственных систем (ГПС). Одним из наиболее эффективных средств создания ГПС, как показала практика, является использование интеллектуальных роботов (ИР), способных принимать решения в соответствии с заложенными в них алгоритмами. Поэтому в отечественной и зарубежной литературе в последнее время уделяется существенное внимание вопросам разработки алгоритмических и программных средств управления поведением ИР как элементов производственной цепи. Не вызывает сомнений справедливость утверждения, что в обозримом будущем если не полностью, то в большой степени ИР должны заменить человека-оператора. Это и составляет одну из основных идей создания ГПС.

Здесь будут рассмотрены как пример ГПС только вопросы принятия решений ИР-сортировщиком по отбраковке изделий на потоке (конвейерном производстве), а также некоторые вопросы «координации» функционирования коллектива ИР.

Основная цель данного раздела — проиллюстрировать применение математического аппарата описания модели мира ИР, использующего набор логических правил *ЕСЛИ... ТО... ИНАЧЕ*, разработанных в моделях нечетких логик (см. раздел 1) [89—115].

Условимся, что у ИР имеются сенсорная система, т. е. датчики для съема тактильной информации, а также исполнительный орган в виде серийного манипулятора. Сам процесс принятия решений реализуется встроенной микроЭВМ и состоит из двух этапов: определения качественной оценки изделия (заготовки) на основе поступающей информации и определения угла поворота манипулятора по полученной качественной оценке заготовки.

Подобный подход, основанный на разработанных правилах нечеткого условного вывода, позволяет, в конечном счете, реализовать управляющие воздействия (например, на серводви-

Значение переменной		$u_i \in U_{x_i}$
«ПАРАМЕТР i »	«КАЧЕСТВО 1»	$v_i \in V_Y$
Несущественное	Предельно низкое	0
Почти малое	Почти низкое	1
Малое	Низкое	2
Чуть больше, чем малое	Чуть лучше, чем низкое	3
Почти среднее	Почти среднее	4
Среднее	Среднее	5
Чуть больше, чем среднее	Чуть лучше, чем среднее	6
Почти большое	Почти высокое	7
Большое	Высокое	8
Чуть больше, чем большое	Чуть лучше, чем высокое	9
Предельное	Наивысшее	10

гатель манипулятора). В результате, как следствие, обеспечивается независимость процедуры отбраковки заготовок (изделий) от субъективных причин, связанных с невысокой квалификацией или степенью утомляемости работников. Одновременно осуществляются повторная обработка той части заготовок, которая ранее браковалась, а также подача высококачественных заготовок для дальнейшей обработки. Поскольку реализация процесса принятия решений ИР основана на использовании правил нечеткого условного вывода, то управление манипулятором-сортировщиком осуществляется на основе определения нечеткого множества качества изделий на потоке. Рассматриваемый подход обеспечивает простоту программной реализации и может быть рекомендован для использования в различных дискретных производствах. Здесь описывается алгоритм принятия решений ИР, реализованный при создании роботизированного участка производства теплообменников на Сумгайтском алюминиевом заводе [82].

Поскольку предполагается наличие у ИР сенсорной системы, то съем тактильной информации будем описывать в виде набора параметров $x = \{x_i\}$, $i = \overline{1, n}$. При этом $x_i \in [x_{i\min}, x_{i\max}]$, $i = \overline{1, n}$, где $x_{i\min}$, $x_{i\max}$ — минимальное и максимальное значения i -того параметра соответственно. Пусть выходная информация о качестве изделия определяется некоторым \tilde{Y} . Параметры из набора $\{x_i\}$ и \tilde{Y} будем рассматривать как нечеткие множества. Более строго, параметры из набора $\{x_i\}$, $i = \overline{1, n}$ рассматриваются как нечеткие множества, формирующие лингвистические переменные, описываемые тройками вида $x_i^0 = \{ \langle x_i^j, U_{x_i}, \tilde{x}_i \rangle \}$. Здесь переменная x_i^j выбирается из терм-множества $T_i^*(u)$ лингвистической переменной «ПАРАМЕТР». Нормальное нечеткое множество \tilde{x}_i описывается функцией принадлежности (ФП) $\mu_{\tilde{x}_i}: U_{x_i} \rightarrow [0, 1]$. При этом U_{x_i} — универсумы вида $U_{x_i} = \{0, 1, 2, \dots, 10\}$.

Приведенной тройке параметров, характеризующей текущую тактильную информацию, ставится в соответствие нечеткое множество качества заготовки Y , формирующего лингвистическую переменную и описываемого тройкой вида $Y^0 = \{ \langle Y_j, V_Y, \tilde{Y} \rangle \}$. Здесь Y_j выбирается из расширенного терм-множества $T^*(\Phi)$ лингвистической переменной «КАЧЕСТВО 1». Нормальное нечеткое множество \tilde{Y} описывается ФП вида $\mu_Y: V_Y \rightarrow [0, 1]$, где V_Y — универсум вида $V_Y = \{0, 1, 2, \dots, 10\}$.

Таким образом, сформированы шкалы оценки параметров текущей тактильной информации $\{x_i\}$, поступающей от сенсорной системы, и качества изделия Y . В табл. 2.26 приведены значения шкалы лингвистических переменных «ПАРАМЕТР i » и «КАЧЕСТВО 1».

Поскольку исполнительным элементом ИР-сортировщика является поворотный манипулятор, то задача принятия решений ИР заключается в определении угла поворота этого манипулятора в зависимости от качества изделия.

Входным параметром для определения угла поворота ИР-сортировщика является новое нечеткое множество \tilde{Y}_1 , формирующее новую лингвистическую переменную «КАЧЕСТВО 2» в виде следующей тройки $Y_1^0 = \{ \langle Y_{1i}, W_{Y_1}, \tilde{Y}_1 \rangle \}$. По аналогии с предыдущим, параметр Y_{1i} выбирается из расширенного терм-множества $T^*(\omega)$ лингвистической переменной «КАЧЕСТВО 2». Нечеткое множество \tilde{Y}_1 описывается ФП вида $\mu_{\tilde{Y}_1}: W_{\tilde{Y}_1} \rightarrow [0, 1]$, где W_{Y_1} — универсум вида $\{0, 1, 2\}$.

В свою очередь, угол поворота манипулятора-сортировщика выражается в виде нечеткого множества \tilde{Z} , которое формирует лингвистическую переменную «УГОЛ» в виде тройки $Z_0 = \{ \langle Z_i, W_z, \tilde{Z} \rangle \}$. Тогда переменная z_i выбирается из расширенного терм-множества $T^*(\omega_z)$ лингвистической переменной «УГОЛ». Нечеткое множество описывается ФП вида $\mu_z: W_z \rightarrow [0, 1]$, где W_z — универсум вида $W_z = \{0, 1, 2\}$. Таким образом, имеем шкалу лингвистических переменных «КАЧЕСТВО 2» и «УГОЛ», которая приведена в табл. 2.27.

Следовательно, необходимо установить отображение лингвистической переменной «КАЧЕСТВО 1», определяемой по 10-балльной шкале табл. 2.26 из текущей тактильной информации, в лингвистическую переменную «КАЧЕСТВО 2», определяемую по 3-балльной шкале табл. 2.27 и используемую при принятии решения в виде поворота манипулятора на заданный угол. Если разбить шкалу табл. 2.26 на три подмножества вида

Значение переменной		$w_i \in W_{U_1}$
«КАЧЕСТВО 2»	«УГОЛЬ»	
Неустранимый дефект Возможна повторная обработка На следующую операцию	Малый	0
	Средний	1
	Большой	2

$V_{1Y} = \{0, 1, 2, 3, 4\}$, $V_{2Y} = \{5, 6, 7\}$, $V_{3Y} = \{8, 9, 10\}$, то для $\{V_{1Y}, V_{2Y}, V_{3Y}\}$ можно поставить в соответствие шкалу из $W_Y = \{0, 1, 2\}$. Таким образом, можно установить отображение $H_1: V_Y \rightarrow W_Y$ в виде: $w^* = 0$ при $v_i^* \in V_{1Y}$; $w^* = 1$ при $v_i^* \in V_{2Y}$ и $w^* = 2$ при $v_i^* \in V_{3Y}$.

Рассмотрим теперь более общий случай, когда задан коллектив ИР.

Пусть роботизированный участок представляет собой совокупность транспортеров $\{k_h\}$, $h=1, 2$, движущихся со скоростью ω_1 и ω_2 соответственно, и поворотных манипуляторов $\{M_j\}$, $j=$

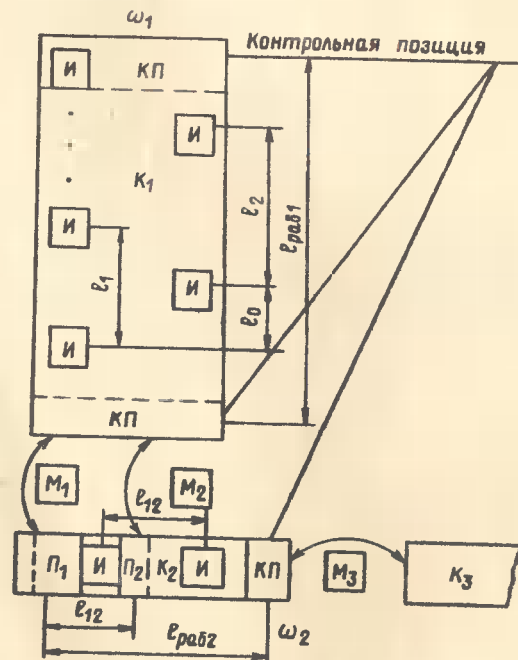


Рис. 20. Фрагмент схемы роботизированного участка. Пояснения в тексте

$= 1, 2, 3$ с постоянным временем срабатывания $\{\tau_j = \text{const}\}$, $j = 1, 2, 3$, расположенных, как показано на рис. 20. Пусть входная информация для манипуляторов, рассматриваемых как ИР, представляется для транспортера k_1 в виде набора расстояний между изделиями $L_1 = \{l_k\}$, $k=0, 1, 2$; $l_k \in [l_{k \min}, l_{k \max}]$, а для k_2 — в виде $L_2 = l_{12} \in [l_{12 \min}, l_{12 \max}]$, где $l_{k \min}$ ($l_{12 \min}$), $l_{k \max}$ ($l_{12 \max}$) — минимально и максимально допустимое расстояние между изделиями на транспортере. При этом $l_{k \min}$ ($l_{12 \min}$), $k=1, 2$ определяется временем срабатывания τ_j , $j=1, 2, 3$ манипуляторов с учетом скорости транспортеров ω_h , $h=1, 2$. В свою очередь, $l_{k \max} = l_{раб1}$ ($l_{12 \max} = l_{раб2}$) $k=0, 1, 2$, где $l_{раб1}$ ($l_{раб2}$) — длина рабочей части транспортера k_1 (k_2). Наборы L_1 и L_2 определяются разницей времени прихода изделий на контрольные позиции и скоростью транспортеров. Отметим, что l_0 характеризуется рассогласованием синхронности поступления каждой пары изделий на транспортер K_1 . Принимается, что $l_0 < 0$, если одно изделие поступает раньше на обслуживание манипулятором M_2 , чем другое — на M_1 ; $l_0 = 0$, если одновременно; $l_0 > 0$, если раньше на M_1 . В последнем случае может возникнуть конфликт, т. е. изделие, обслуживаемое манипулятором M_1 , двигаясь на транспортере K_2 , попадает на рабочую позицию Π_2 в тот момент, когда M_2 обслуживает другое изделие, что технологически недопустимо. В этой связи строится следующая шкала для l_0 :

$$l_{0 \max} = \frac{(l_{12}^0 + l_{12 \min})}{\omega_2} \omega_1, \quad l_{0 \min} = \frac{(l_{12}^0 - l_{12 \min})}{\omega_2} \omega_1,$$

где l_{12}^0 — расстояние между рабочими позициями Π_1 и Π_2 (см. рис. 20). Пусть выходная информация определяется временем задержки срабатывания манипуляторов $\tau_{зад j}$, $j=1, 2, 3$. Таким образом, истинное время срабатывания определяется как $\tau_{оп j} = \tau_j + \tau_{зад j}$, $j=1, 2, 3$. При этом $\tau_{зад j} \in [0, \tau_{зад \max}]$, где

$$\tau_{зад \max} = \begin{cases} l_{раб h} / \omega_h, & h=1, 2, l_0 \leq 0, \\ 2l_{12 \min} / \omega_2, & l_0 > 0. \end{cases}$$

Далее параметры входной и выходной информации рассматриваются как нечеткие множества, формирующие лингвистические переменные, описываемые тройками вида $L_k^0 = \{ \langle L_k^j, W_L, \bar{L}_k \rangle \}$, $L_k^j \in T_k^*(W_L)$ (\bar{L}_3 соответствует l_{12}), $k=0, 1, 2, 3$, $j=0, 1, \dots, 10$, $\tau_{зад j}^m = \{ \langle \tau_{зад j}^m, W_\tau, \bar{\tau}_{зад j} \rangle \}$, $\tau_{зад j}^m \in T_j^*(W_\tau)$, $j=1, 2, 3$, $m=0, 1, \dots, 10$, где $T_k^*(W_L)$ — соответственно терм-множества лингвистических переменных «РАССТОЯНИЕ К» и «ЗАДЕРЖКА j»; \bar{L}_k , $\bar{\tau}_{зад j}$ — нормальные нечеткие множества с ФП вида $\mu_{\bar{L}}: W_L \rightarrow [0, 1]$, $\mu_{\bar{\tau}}: W_\tau \rightarrow [0, 1]$; $W_L = W_\tau = \{0, 1, 2, \dots, 10\}$ — универсумы. Лингвистические переменные «РАССТОЯНИЕ К» и «ЗАДЕРЖКА j» принимают значения из набора лингвистических термов вида: малое(ая), ..., среднее(ая), ..., ..., большое(ая).

Для описания модели окружающего мира ИР предлагаются правила нечеткого условного вывода в виде следующих бинарных отношений.

Для ИР-сортировщика ($\forall i=1, 2, \dots, n$)

$$R_1(A_1(x), A_2(y)) = \left(\left[\bigcap_i \tilde{X}_i \right] \times V_{Y \rightarrow U_{X_i}} \times \tilde{Y} \right) \cap \left(\bigcap_i \left[\bigcap_i \tilde{X}_i \right] \times V_{Y \rightarrow U_{X_i}} \times \neg \tilde{Y} \right), \quad (2.44)$$

где \times — декартово произведение, \cap — пересечение множеств, \rightarrow — операция импликации используемого варианта нечеткой логики (см. раздел 1), \neg — знак инверсии, а также

$$R_2(A_1(y_1), A_2(z)) = (\tilde{Y}_1 \times W_{z \rightarrow W_{Y_1}} \times \tilde{z}) \cap \left(\neg \tilde{Y}_1 \times W_{z \rightarrow W_{Y_1}} \times \neg \tilde{z} \right), \quad (2.45)$$

где \rightarrow — операция импликации в S -логике (см. раздел 1).

Для «координации» коллектива ИР

$$R_{3d}(A_1(l_k), A_2(\tau_{задj})) = (\tilde{L}_k \times W_{\tau \rightarrow W_L} \times \tilde{\tau}_{задj}) \cap \left(\neg \tilde{L}_k \times W_{\tau \rightarrow W_L} \times \neg \tilde{\tau}_{задj} \right), \quad d=1, 2, 3, 4. \quad (2.46)$$

В терминах ФП выражения (2.44), (2.45) принимают вид

$$R_1(A_1(x), A_2(y)) = \int_{U_{X_i} \times V_Y} \left(\left(\bigwedge_i \mu_i(u) \right) \rightarrow \mu(v) \right) \wedge \wedge \left[1 - \left(\bigwedge_i \mu_i(u) \right) \rightarrow (1 - \mu(v)) \right] (u, v) = \begin{cases} 1 - \mu(v), & \left(\bigwedge_i \mu_i(u) \right) < \mu(v), \\ 1, & \left(\bigwedge_i \mu_i(u) \right) = \mu(v), \\ \mu(v), & \left(\bigwedge_i \mu_i(u) \right) > \mu(v), \quad \forall i=1, 2, \dots, n; \end{cases} \quad (2.47)$$

$$R_2(A_1(y_1), A_2(z)) = \begin{cases} 1, & \mu(W_{Y_1}) = \mu(W_z), \\ 0, & \mu(W_{Y_1}) \neq \mu(W_z). \end{cases}$$

Для бинарных отношений вида (2.46) получаются выражения, аналогичные (2.47), т. е. набор

$R_{31}(A_1(l_1), A_2(\tau_{зад1}))$, $R_{32}(A_1(l_2), A_2(\tau_{зад2}))$, $R_{33}(A_1(l_3), A_2(\tau_{зад3}))$, $R_{34}(A_1(l_4), A_2(\tau_{зад4}))$, а для формирования выражений типа (2.44) используются сформулированные экспертным путем условные предложения вида:

«если \tilde{X}_1 есть A_1 и ... и \tilde{X}_n есть A_n , то \tilde{Y} есть B , иначе \tilde{Y} есть не B ».

где $A_i \subseteq U_{X_i}$, $i=1, 2, \dots, n$, $B \subseteq V_Y$ — нечеткие множества, соответствующие значениям лингвистических переменных «ПАРАМЕТР i » и «КАЧЕСТВО 1» из набора термов, описанных в табл. 2.26. Непосредственно процесс принятия решений ИР-сортировщиком заключается в определении нечеткого множества качества изделия $\tilde{Y}_{тек}$, соответствующего нечетким множествам текущих значений набора входных параметров $\{\tilde{X}_i^{тек}\}$, затем в соответствии с $\tilde{Y}_1^{тек}$ — в определении нечеткого множества $\tilde{Z}^{тек}$. Для этой цели используется идея «композиционного вывода» Л. Заде [6], т. е.

$$\tilde{Y}_{тек} = \left[\bigcap_i \tilde{X}_i^{тек} \right] \circ R_1(A_1(x), A_2(y)), \quad i=1, 2, \dots, n,$$

и далее $\tilde{Z}^{тек} = \tilde{Y}_1^{тек} \circ R_2(A_1(y_1), A_2(z))$, где \circ — операция максимальной композиции; $R_1(\cdot)$, $R_2(\cdot)$ — бинарные отношения вида (2.44) и (2.45) соответственно. Далее отметим, что

$$\mu_{W_z^*} \in W_z \mid \mu_z(w_z^*) = \sup_{w_z} \mu_z(w_z), \quad j=1, 2, 3,$$

где $\mu_z(w_z)$ — оценка ФП для j -го синглтона. Подставляя w_z^* в соотношение вида $W_{zj} = \text{Ent}[(\text{Card}(W_z) - 1) \cdot (\varphi_j - 90) / 180]$ и учитывая, что мощность множества W_z типа $\text{Card } W_z = 3$, получим выражение для определения угла поворота манипулятора $\varphi_j = \frac{\pi}{2}(W_z^* + 1)$. При этом управление манипулятором может осуществляться в режиме непосредственного цифрового управления с микроЭВМ. Та же идея «композиционного вывода» используется и при принятии решений по «координации». При $l_0 \leq 0$, т. е. если «конфликта» нет, задержки времени срабатывания манипуляторов, используемые для адаптации темпа работы манипуляторов к интенсивности поступления изделий, определяются как $\tilde{\tau}_{задj}^{тек} = \tilde{L}_k^{тек} \circ R_{3d}(A_1(l_k), A_2(\tau_{задj}))$, $j=k=d=1, 2, 3$, а в случае «конфликта», т. е. при $l_0 > 0$, функции «координатора» выполняет манипулятор M_2 . При этом $\tilde{\tau}_{зад2}^{тек} = \tilde{L}_0^{тек} \circ R_{34}(A_1(l_0), A_2(\tau_{зад2}))$.

Рассмотренный ИР-сортировщик был испытан на потоке для определения качества отшлифованных испарителей в ГПС Сумгайтского алюминиевого завода. Здесь ограничимся приведенным качественным описанием работы ИР-сортировщика. Более подробные характеристики описанного алгоритма принятия решений ИР подобного типа даны в [82].

ЗАКЛЮЧЕНИЕ

В данном обзоре дается описание теории нечетких моделей динамических объектов и внедренных в практику нечетких алгоритмов и систем автоматического управления, обладающих

следующими преимуществами: 1) простотой структуры управляющей программы и быстрой сходимостью к оптимальному режиму; 2) минимальным объемом необходимой памяти, что дает возможность реализовать алгоритм на микропроцессорах; 3) устойчивостью функционирования (нечувствительностью, грубостью и инвариантностью) в оптимальном режиме.

Перечисленные преимущества являются часто одним из основных требований при проектировании САУ и учитываются при разработке соответствующих баз знаний и данных САПР или персональных интеллектуальных АРМ разработчика. При описании конкретных нечетких моделей систем управления [2, 7, 9, 19, 31, 106—115] особое внимание уделяется вопросам применения в ГПС, роботизированных производствах и автономных интеллектуальных роботах-манипуляторах, в адаптивных САУ сложными динамическими системами (различного типа летательные аппараты, вспомогательные биотехнические аппараты замещения утраченных функций органов и др.), при разработке баз знаний и экспертных систем, САПР, интеллектуального или логического программирования, в оценке надежности и принятия решений в экстремальных условиях с учетом работы человека-оператора и др.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ ИНФОРМАЦИИ

1. Петров Б. Н., Уланов Г. М., Ульянов С. В., Хазен Э. М. Информационно-семантические проблемы в процессах управления и организации.— М.: Наука, 1977.— 452 с.
2. Петров Б. Н., Уланов Г. М., Гольденблат И. И., Ульянов С. В. Теория моделей в процессах управления (информационный и термодинамический аспекты).— М.: Наука, 1978.— 224 с.
3. Petrov B. N., Pugachev V. S., Ulanov G. M., Krinetsky O. E., Ulyanov S. V. Informational foundations of qualitative theory of control systems // Proc. VII Congr. IFAC—Finland: Academic Press, 1978.— 3.— P. 1763—1769.
4. Петров Б. Н., Уланов Г. М., Ульянов С. В. Динамические системы со случайной и нечеткой структурами // Итоги науки и техники. Сер. Техн. кибернетика.— М.: ВИНТИ, 1979.— Т. 11.— С. 3—76.
5. Петров Б. Н., Уланов Г. М., Гольденблат И. И., Ульянов С. В. и др. Информационные аспекты качественной теории динамических систем // Итоги науки и техники. Сер. Техн. кибернетика.— М.: ВИНТИ, 1976.— Т. 7.— С. 5—201; 1977.— Т. 8.— С. 102—208; 1978.— Т. 10.— С. 5—119.
6. Заде Л. А. Понятие лингвистической переменной и его применение к принятию приближенных решений.— М.: Мир, 1976.— 165 с.
7. Kaufmann A., Gupta M. M. Fuzzy mathematical model with applications to engineering and management science.— Amsterdam, North Holland: Elsevier, 1988.— 376 p.
8. Кофман А. Введение в теорию нечетких множеств.— М.: Радио и связь, 1982.— 432 с.
9. Evans G. W., Karwowsky W., Wilhelm M. R. (Eds.) Applications of fuzzy sets methodologies in industrial engineering. Advances in industrial engineering. V. 8.— Amsterdam, North Holland: Elsevier, 1989.— 335 p.
10. Борисов А. Н., Алексеев А. В., Крумберг О. А., Меркурьева Г. В., По-

- пов В. А. Модели принятия решений на основе лингвистической переменной.— Рига: Зинатне, 1982.— 256 с.
11. Zadeh L. A. Knowledge representations in fuzzy logic // IEEE Trans on Knowl. and Data Eng.— 1989.— 1, № 1.— P. 89—100.
 12. Gupta M. M., Yamaoka T. (Eds.). Fuzzy logic in knowledge-based systems.— Amsterdam, North Holland: Elsevier, 1988.— 376 p.
 13. Поспелов Д. А. Логико-лингвистические модели в системах управления.— М.: Энергоиздат, 1984.— 232 с.
 14. Кузьмин В. Б. Построение групповых решений в пространстве четких и нечетких бинарных отношений.— М.: Наука, 1982.— 168 с.
 15. Zadeh L. A., Fu K. S., Tanaka K., Mizumoto M. (Eds.). Fuzzy sets and their applications to cognitive and decision processes.— N. Y.: Academic Press, 1975.— 496 p.
 16. Negoita C. V., Ralescu D. A. Applications of fuzzy sets to systems analysis.— Basel: Birkhäuser Verlag, 1975.— 190 p.
 17. Dubois D., Prade H. Fuzzy sets and systems: Theory and applications.— N. Y.: Academic Press, 1980.— 342 p.
 18. Нечеткие множества и теория возможностей. Последние достижения / Пер с англ. под ред. П. Р. Ягера.— М.: Радио и связь, 1986 г.— 408 с.
 19. Аверкин А. Н., Батыршин И. З., Блишун А. Ф., Силов В. Б., Тарасов В. Б. Нечеткие множества в моделях управления и искусственного интеллекта.— М.: Наука, 1986.— 312 с.
 20. Kania A. A., Kiszka J. B., Gorzalczyk M. B., Maj T. R., Stachowicz M. S. On stability of formal fuzziness systems // Inf. Sci.— 1980.— 22, № 1.— P. 51—68.
 21. Kiszka J. B., Kochańska M. E., Słwińska D. S. The influence of some implication operators on the accuracy of a fuzzy model. Pt. I, II // Int. J. Fuzzy Sets and Systems.— 1985.— 15, № 2.— P. 111—128.
 22. Gupta M. M., Trojan G. M., Kiszka J. B. Controllability of fuzzy control systems // IEEE Trans. Syst. Man, and Cybern.— 1986.— SMC-16, № 4.— P. 577—582.
 23. Gupta M. M., Kiszka J. B., Trojan G. M. Multivariable structure of fuzzy control systems // IEEE Trans. Syst. Man and Cybern.— 1986.— SMC-16, № 5.— P. 628—656.
 24. Xu Chen-wei, Lu Young-zai. Decoupling in fuzzy systems: a cascade compensation approach // Int. J. Fuzzy Sets and Syst.— 1989.— 29, № 2.— P. 177—185.
 25. Kawase S., Yanagihara N. On the stability of fuzzy control systems // Proc. III Cong. on Fuzzy Systems Siettle, 15—23 Jun.— Siettle, USA, 1989.— P. 67—70.
 26. Wakami N., Akahori H., Ichihashi H. A comparative study of fuzzy sequence method control and linear feedback control // Ibid.— P.63—66.
 27. Swietniak A. Fuzzy controllers design via fixed point theorems // Int. J. Fuzzy Sets and Syst.— 1986.— 20, № 2.— P. 131—136.
 28. Нариньяни А. С. Недоопределенность в системе представления и обработки знаний // Изв. АН СССР. Техн. кибернетика.— 1986.— № 5.— С. 3—25.
 29. Mizumoto M., Zimmerman H. J. Comparison of fuzzy reasoning methods // Int. J. Fuzzy Sets and Syst. 1982.— 8, № 3.— P. 253—283.
 30. Zadeh L. A. Yes, no and relatively... Pts. I, II // Chemitech.— 1987.— 17, № 7.— P. 406—410.
 31. Sugeno M. Industrial applications of fuzzy systems.— North Holland: TÜV, 1985.— 431 p.
 32. Gaines B. R. Fuzzy and probability uncertainty logic // Int. and Contr.— 1978.— 38, № 2.— P. 154—169.
 33. Bellman R. E., Ladeh L. A. Local and fuzzy logic // Memorandum ERL—M584.— Berkeley: Electronic Research Laboratory. College of Eng. University of California, 1976.— 83 p.
 34. Калужнин Л. А. Введение в общую алгебру.— М.: Наука, 1973.— 448 с.

35. Федерер Г. Геометрическая теория меры.— М.: Мир, 1987.— 760 с.
36. Клини С. Математическая логика.— М.: Мир, 1973.— 480 с.
37. Непейвода Н. Н. Логические и алгоритмические формализмы для задачи построения правильных программ // Программирование.— 1981.— № 6.— С. 3—14.
38. Непейвода Н. Н. Конструктивные логические средства. I. Обобщенное понятие логического исчисления // Изв. АН СССР. Техн. кибернет.— 1987.— № 5.— С. 64—74.
39. Ефимов Е. И. Дедуктивные рассуждения с нечеткими кванторами // Изв. АН СССР. Техн. кибернетика.— 1987.— № 5.— С. 120—133.
40. Baldwin J. F., Pilsworth B. Axiomatic approach to implication for approximate reasoning with fuzzy logic // Int. J. Fuzzy Sets and Syst.— 1980.— 3, № 2.— P. 193—219.
41. Bandler W., Kohout L. Fuzzy power sets and fuzzy implications operators // Ibid.— 1980.— 4, № 1.— P. 13—30.
42. Okamoto M. B. A measure of closeness of weak implication for strict implication // Proc. Int. Conf. Artif. Intell. Tokyo, 18—25 Aug.— Tokyo, 1979.— P. 96—106.
43. Поспелов Г. С., Поспелов Д. А. Искусственный интеллект. Прикладные системы.— М.: Знание, 1985.— 34 с.
44. Рахманова З. Б., Ульянов С. В. Экспертная система для лечебно-диагностических процессов // Мед. техника.— 1988.— № 6.— С. 23—41.
45. Вагис А. Г., Братусь А. В., Василенко В. И. Имитационное моделирование систем на основе логических выводов // Докл. АН УССР, Сер. А: Физ.-мат. и техн. науки, 1987.— № 10.— С. 62—64.
46. Schwartz D. G. Axiom for a theory of semantic equivalence // Int. J. Fuzzy Sets and Syst.— 1987.— 21, № 3.— P. 319—349.
47. Ульянов С. В. Стохастические и нечеткие модели релятивистских и квантовых динамических систем: физические особенности объектов управления // Нечеткие системы поддержки принятия решений. КГУ.— 1989.— С. 15—28.
48. Ульянов С. В. Модели квантово-релятивистских нечетких логик в интеллектуальных системах // Тр. II Всесоюз. Съезда по искусственному интеллекту.— Минск: Минский Гос. ун-т, 1990.— С. 81—86.
49. Петров Б. Н., Уланов Г. М., Ульянов С. В. Сложность конечных объектов и информационная теория управления // Итоги науки и техники. Сер. Техн. кибернетика.— М.: ВИНТИ, 1979.— Т. 11.— С. 77—147.
50. Rescher N. Many valued logic. N. Y.: McGraw Hill. 1969.— 312 p.
51. Mamdani E. N., Sembli B. S. On the nature of implication in fuzzy logic // Proc. 9th Int. Symp. Multiple-Valued Logics.— Bath, 1979. New York, N. Y. 1979.— P. 143—151.
52. Церковный А. Э. Нечеткая логика и приближенные рассуждения // Нечеткие системы: моделирование структуры и оптимизация.— Калинин: Калинин. гос. ун-т, 1987.— С. 31—39.
53. Петров Б. Н., Гольденблат И. И., Уланов Г. М., Ульянов С. В. Проблемы управления релятивистскими и квантовыми динамическими системами.— М.: Наука, 1982.— 525 с.
54. Ali S. T., Prugovečki E. Systems of imprivity and representations of quantum mechanics on fuzzy phase space // J. Math. Phys.— 1977.— 18, № 12.— P. 219—228.
55. Bush P., Lahti P. J. Minimal uncertainty and maximal information for quantum position and momentum // J. Phys.— 1987.— 20A, № 11.— P. 899—906.
56. Mittelstaedt P. EPR-paradox, quantum logic and relativity // Symp. Found. Mod. Phys. // 50 Years of the Einstein—Podolsky—Rosen Gedanken-experiment.— Singapore, World Scientific Publ. Co., 1985.— P. 171—186.
57. Schroek E. E. On the nonoccurrence of two paradoxes in the measurement scheme of stochastic quantum mechanics // Found. Phys.— 1985.— 15, № 3.— P. 279—302.
58. Zurek W. H. Environment-induced superselection rules // Phys. Rev.— 1982.— 26D, № 8.— P. 1862—1880.
59. Grössing G. Quantum cybernetics and its test in «late choice» experiments // Phys. Lett.— 1986.— 118A, № 8.— P. 381—386.
60. Hsu J. P. Four-dimensional symmetry from a broad viewpoint. VIII. Fuzzy quantum mechanics and evolutionary laws of physics // Nuovo Cimento.— 1985.— 89B, № 1.— P. 14—29.
61. Jannusiss A., Papaloucas L., Tsilimigras P., Tsoulis K. The R-inherent quantum mechanics of Hsu as a particular case of the Lie-isotopic lifting theory // Hadron J. Suppl.— 1986.— 2, № 3.— P. 498—508.
62. Кандрашина Е. Ю., Литвинцева Л. В., Поспелов Д. А. Представление знаний о времени и пространстве в интеллектуальных системах.— М.: Наука, 1989.— 328 с.
63. Stachow E. W. Structures of quantum language for compound systems // Symp. Found. Mod. Phys. / 50 Years of the Einstein—Podolsky—Rosen Gedankenexperiment.— Singapore, World Scientific Publ. Co., 1985.— P. 625—635.
64. Yumarie G. Subjectivity, information, systems: introduction to a theory of a relativistic cybernetics.— Gordon and Breach. N. Y. 1986.— 326 p.
65. Prugovečki E. Stochastic quantum mechanics and quantum spacetime: a consistent of relativity and quantum theory of stochastic spacetime // Foundational Theory in Physics—Dordrecht et al.: Reidel, 1984.— 302 p.
66. Guz W., Fuzzy σ -algebras of physics // Int. J. Theor. Phys.— 1985.— 24, № 5.— P. 481—493.
67. Ligomenides P. A. Symbolic space determinations on physical limitations // Ibid.— 1982.— 21, № 12.— P. 973—978.
68. Фейнман Р. Ф. Квантово-механические ЭВМ / Успехи физ. наук.— 1986.— 149, Вып. 4.— С. 671—688.
69. Yamakawa T. Intrinsic fuzzy electronic circuits for sixth generation computers // Fuzzy computing: theory, hardware, and applications. // Gupta M. M., Yamakawa T. (Eds.).— North Holland, Amsterdam. New York et al.— 1988.— P. 157—172.
70. Kickert W. J., van Lemke N. The application of fuzzy set theory to control a warm water plant // Automatica.— 1976.— 12, № 3.— P. 301—308.
71. Mamdani E. H. Applications of fuzzy sets theory to control systems: A survey // Fuzzy automata and decision processes // Gupta M. M. et al. (Eds.).— North-Holland, Amsterdam, New York et al.— 1979.— P. 77—88.
72. Mamdani E. H., Ostergaard J. J., Lembesis E. Use of fuzzy logic for implementing rule-based control of industrial processes // Fuzzy Sets and Decision Making / Janchez E., Gupta M. M. (Eds.).— North-Holland, Amsterdam, New York et al.— 1984.— P. 48—62.
73. Chang S. S. L., Zadeh L. A. On fuzzy mapping and control // IEEE Trans. Syst., Man, and Cybern.— 1972.— SMC-2, № 1.— P. 30—34.
74. Tong R. M. A retrospective view of fuzzy control systems // Fuzzy Sets and Syst.— 1984.— 14, № 3.— P. 199—210.
75. Sugeno M. On application of fuzzy sets and fuzzy logic to control // J. Society of Instr. and Contr. Eng.— 1979.— 18, № 2.— P. 150—160 (on jap.).
76. Aliev R. A. Production control on the basic of fuzzy models // Int. J. Fuzzy Sets and Syst.— 1987.— 22, № 1.— P. 43—56.
77. Ray K. S., Hajjamber O. Fuzzy logic controller of a nonlinear multivariable steam generating unit using decoupling theory // IEEE Trans. Syst., Man, and Cybern.— 1985.— SMC-15, № 4.— P. 539—558.
78. Mamdani E. H. Advances in the linguistic synthesis of fuzzy controllers // Int. J. Man-Mach. Stud.— 1976.— 8, № 6.— P. 669—678.
79. Mamdani E. H. Rule-based fuzzy approach to the control of dynamic processes // IEEE Trans. on Computers.— 1979.— 12, № 3.— P. 432—440.
80. Алиев Р. А., Джафаров С. М., Гасанов И. С., Бабаев М. Д., Быстродействующий алгоритм композиционного вывода и его применение в раз-

мытых регуляторов // Изв. ВУЗов. Нефть и газ.— 1985.— № 11.— С. 80—84.

81. Алиев Р. А., Джафаров С. М., Бабаев М. Д. Конструирование нечеткого адаптивного регулятора // Деп. ВИНТИ.— 1986.— № 11(181).— 18 с.
82. Алиев Р. А., Церковский А. Э. Интеллектуальный робот для оценки качества и сортировки изделий на потоке // Изв. АН СССР. Техн. киберн. 1986.— № 1.— С. 100—106.
83. Васильева О. И., Ионов И. П., Кантор П. С., Ульянов С. В. Дуальное управление процессом искусственной вентиляции легких с использованием нечеткого регулятора в цепи обратной связи // Мед. техника.— 1989.— № 1.— С. 11—21.
84. Петров Б. Н. Избранные труды. Т. 1. Т. 2.— М.: Наука, 1983.— 430 с., 327 с.
85. Mamdani E. H., Assilian S. An experiments in linguistic synthesis with a fuzzy logic controller // Int. J. Man-Machine Studies.— 1975.— 7, № 1, P. 1—13.
86. Procyk T. J., Mamdani E. H. A linguistic self-organizing process controller // Automatica.— 1979.— 15, № 1.— P. 15—30.
87. Assilian S. Artificial intelligence in the control of real dynamic systems // Th. Doct. Dis.— Augustin, 1974.— 23 p.
88. Braae M., Rutherford D. A. Theoretical and linguistic aspects of the fuzzy logic controller // Automation.— 1979.— 12, № 9.— P. 553—557.
89. Schwartz D. G. Fuzzy inference in a formal theory of semantic equivalence // Int. J. Fuzzy Sets and Syst.— 1989.— 31, № 2.— P. 205—216.
90. Ahlgvist J. E. Application of fuzzy implication to probe nonsymmetric relations' pts I, II // Ibid.— 1988.— 25, № 1.— P. 87—95.
91. Yager R. R. On the implication operator in fuzzy logic // Inf. Sci.— 1983.— 31, № 2.— P. 141—164.
92. Cao Z., Kandel A. Applicability of some fuzzy implication operators // Int. J. Fuzzy Sets and Systems.— 1989.— 31, № 2.— P. 151—186.
93. Dubois D., Martin-Clonaire R., Prade H. Practical computing in fuzzy logic // Fuzzy Comput. / Eds. by M. M. Gupta, Yamakawa T.— North Holland: Elsevier Science Publ. B. V. — 1988.— P. 11—34.
94. Dubois D., Prade H. On fuzzy syllogisms // Comput. Intell.— Canada, Montreal.— 1987. P. 1—21.
95. Oh Kyung W., Bandler W. Properties of fuzzy implications operators // Int. J. Approxim. Reason.— 1987.— 1, № 3.— P. 273—285.
96. Cook G. W. Distributions in fuzzy logic // Int. J. Fuzzy Sets and Syst.— 1986.— 20, № 3.— P. 355—364.
97. Smets P., Magres P. Implications in fuzzy logic // Int. J. Approxim. Reason.— 1987.— 1, № 4.— P. 327—347.
98. Trillas E., Valverde L. On mode and implication in approximate reasoning // Approximate Reasoning in Expert System / Eds. Gupta M. M. et ul.— North Holland: Elsevier Science Publ. B. V., 1985.— P. 157—166.
99. Hayashi Y., Tazaki E., Yoshida K., Dey F. Medical diagnosis using simplified multi-dimensional fuzzy reasoning // Proc. Ing. in Medicine and Biology Society. / 10th Ann. Int. Conf. New Orleans, Nov. 4—7, 1988.— N. Y., 1988.— Pt. 3/4.— P. 58—62.
100. Hayashi Y., Nakai M. An efficient algorithm for multi-dimensional fuzzy reasoning and its application to fault diagnosis // Trans. Inst. Electr. Eng. Jap. E. C.— 1989.— 109, № 9.— P. 653—660.
101. Hayashi Y., Nakai M. Reasoning method using a fuzzy production rule with linguistic relative importance in an antecedent // Ibid.— P. 661—668.
102. Lin Xu-Hua, Fang Kwang-Ya, Tsai Jing-Pha. The interpretation of fuzzy reasoning in operator fuzzy logic // Proc. 19th Int. Symp. Multiple-Valued Logic.— Guangzhou. 1989.— Washington (D. C.), 1989.— P. 242—247.
103. Cheng L., Chen H. Fuzzy reasoning in the inverse of fuzzy implication // Ibid.— P. 252—254.
104. Cao Z., Kandel A., Ward L. D. Flexible implication operators in fuzzy reasoning // Ibid.— P. 26—30.

105. Piš P., Mesiar R. Fuzzy model of inexact reasoning in medicine // Comput. Meth. and Programs Biomed.— 1989.— 30, № 1.— P. 1—8.
106. Kandel A. Fuzzy mathematical techniques with applications.— Addison-Wesley, 1986.— 312 p.
107. Kaufmann A., Gupta M. M. Introduction to fuzzy arithmetic: theory and application.— Van Nostrand Reinhold, 1985.— 412 p.
108. Daley S., Gill K. F. Attitude control of a spacecraft using an extended self-organising fuzzy logic controller // Proc. Int. Mech. Eng.— 1987.— 201, № C2— P. 97—106.
109. Daley S., Gill K. F. Comprasion of fuzzy logic controller with a P+D control law // Trans. ASME. J. Dyn. Syst. Meas. and Contr.— 1989.— 111, № 2.— P. 128—137.
110. Buckley Y., Ying H. Fuzzy controller theory: limit theorem for linear fuzzy control rules // Automatica. 1989.— 23, № 3.— P. 469—472.
111. Nishikawa T. Fuzzy theory: the science of human intuition // Jap. Comput. Quart.— 1989.— № 79.— P. 25—37.
112. Recently publicized examples of neurocomputer and fuzzy computer development projects // Jap. Comput. Quart.— 1989.— № 79.— P. 38—40.
113. Arikawa H. Fuzzy development station. // Кэйсоку гидзюку (Instrum. and Autom.) 1989.— 17, № 9.— P. 52—56.
114. Hirota K., Ozawa K. The concept of fuzzy flip-flop // IEEE Trans. Syst. Man, and Cybern.— 1989.— 19, № 5.— P. 980—997.
115. Dettloff W. D., Yount K. E., Watanabe H. A VLSI fuzzy logic inference engine for real-time process control // Proc. IEEE Custom. Integr. Circuits Conf., San Diego, Calif., 15—18 May, 1989.— New York, 1989.— P. 1241—1245.

СОДЕРЖАНИЕ

Я. З. Цыпкин, А. С. Позняк, С. Н. Тихонов. Оптимальные методы адаптивной идентификации	3
Введение	4
1. Задача идентификации	13
2. Алгоритмы идентификации	40
Заключение	41
Список использованных источников информации	41
И. В. Нейманн, М. В. Шнейдерман. Получение экспертных знаний	44
Введение	45
1. Проблема получения экспертных знаний	49
2. Основные области получения знаний	55
3. Индивидуальное получение знаний	59
4. Коллективное получение знаний	68
5. Формы получаемых знаний	74
Заключение	76
Список использованных источников информации	76
А. Д. Плитман, М. И. Рубинштейн. Эвристические методы в календарном планировании	79
Введение	80
1. Общая формулировка и классификация задач календарного планирования	83
2. Приближенные методы решения детерминированных и хорошо формализованных задач календарного планирования	91
3. Календарное планирование на базе имитационных моделей	94
4. Календарное планирование на базе методов искусственного интеллекта	94
4.1. Проблема планирования с позиций искусственного интеллекта	95
4.2. Автоматические системы построения планов	102
4.3. Принципы построения экспертных систем и представления знаний	107
4.4. Реализация экспертных систем для календарного планирования	123
Заключение	123
Список использованных источников информации	123
Р. А. Алиев, Э. Г. Захарова, С. В. Ульянов. Нечеткие модели управления динамическими системами	127
Введение	131
1. Нечеткая логика, лингвистическая аппроксимация и нечеткие алгоритмы управления	146
2. Нечеткие модели динамических систем управления	146
2.1. Принципы построения нечетких регуляторов	154
2.2. Аналитические методы синтеза нечетких регуляторов	158
2.3. Лингвистический синтез динамической системы управления	172
2.4. Адаптивный нечеткий регулятор	177
2.5. Примеры внедренных нечетких промышленных динамических систем управления	177
2.5.1. Нечеткая адаптивная система управления ректификационной колонной	177
2.5.2. Нечеткие модели принятия решений в интеллектуальных роботах	189
Заключение	195
Список использованных источников информации	196

УДК 681.51.015

Я. З. Цыпкин, А. С. Позняк, С. Н. Тихонов. Оптимальные методы адаптивной идентификации // Итоги науки и техники. Сер. Техническая кибернетика. — М.: ВИНТИ, 1990. — Т. 29. — С. 3—44.

Обзор посвящен задаче идентификации динамических объектов, описываемых линейными разностными уравнениями известного порядка, в режиме их нормальной работы. Обсуждаются оптимальные методы идентификации, которые включают в себя синтез оптимальных настраиваемых моделей (предикторов), выбор оптимальных функций потерь и синтез оптимальных алгоритмов настройки параметров предикторов. Оптимальные методы идентификации основаны на учете имеющейся априорной информации об идентифицируемом объекте и помехе. Рассматриваются минимаксные на классе, или робастные версии алгоритмов, содержащих нелинейное преобразование обобщенной невязки, которое ориентировано на наилучшую в данном классе помеховую ситуацию. Быстродействие таких процедур сравнивается с предельно возможным путем установления факта асимптотической нормальности нормированных векторов уклонений оценок параметров от истинных значений. Для упрощенных алгоритмов приводятся условия их работоспособности и устанавливается скорость сходимости. На основе описанного подхода проводится сравнение как широко известных алгоритмов адаптивной идентификации, так и новых процедур. Библ. 86.

УДК 65.012.16

И. В. Нейманн, М. В. Шнейдерман. Получение экспертных знаний // Итоги науки и техники. Сер. Техническая кибернетика. — М.: ВИНТИ, 1990. — Т. 29. — С. 44—78

Обзор посвящен изучению проблемы получения экспертных знаний, которая стала в последнее время предметом всеобщего внимания. Актуальность этой проблемы прежде всего связана с разработкой экспертных систем. Как известно, в центре разработки лежит процесс извлечения знаний из экспертов и преобразования их в машинную программу. Однако проблема получения знаний выходит за рамки экспертных систем и проявляется в различных областях. Так, другой важной областью являются методы сбора и анализа экспертной информации, т. е. методы проведения реальных (очных) экспертиз. В работе изучаются общие особенности, этапы, трудности получения знаний. Целенаправленно освещаются методы получения знаний, которые делятся на две группы: индивидуальное и коллективное получение знаний. Обсуждаются вопросы связи методов получения знаний с формами организации знаний у человека.

УДК 007:683.3.06:519.85

А. Д. Плитман, М. И. Рубинштейн. Эвристические методы в календарном планировании. // Итоги науки и техники. Сер. Техническая кибернетика. — М.: ВИНТИ, 1990. — Т. 29. С. 79—127

Обзор посвящен основным направлениям применения эвристических методов в календарном планировании. Первое направление охватывает методы, основанные на анализе формальных моделей реальных задач календарного планирования, т. е. применимые к таким классам задач, для которых могут быть построены, обобщимые, поддающиеся анализу модели. К второму направлению относятся процедуры планирования, опирающиеся на применение имитационных моделей планируемых процессов. В рамках третьего направления представлены методы календарного планирования, основанные на знаниях и реализуемые как в автоматических планировщиках, так и, в большей степени, в экспертных системах. Библ. 97.

УДК 681.51

Р. А. Алиев, Э. Г. Захарова, С. В. Ульянов. Нечеткие модели управления динамическими системами // Итоги науки и техники. Сер. Техн. кибернетика. — М.: ВИНТИ, 1990. — Т. 29. — С. 127—201

Обзор посвящен вопросам анализа и синтеза нечетких моделей управления сложными динамическими системами. Основное внимание уделяется прикладным аспектам теории нечетких моделей алгоритмов управления в САУ. Приводятся оценки сложности внедренных алгоритмов управления и сравнительные оценки с другими известными алгоритмами управления. Описание и анализ работоспособности внедренных алгоритмов и систем управления на базе нечеткой логики (лингвистических переменных) позволяет глубже оценить целесообразность, предельные возможности используемых подходов и содержательную интерпретацию результатов теории нечетких моделей. Библ. 115.