

## АЛГОРИТМ ПЕРЕМЕЩЕНИЯ МАНИПУЛЯЦИОННОГО РОБОТА В СРЕДЕ С НЕИЗВЕСТНЫМИ ПРЕПЯТСТВИЯМИ

*Рассматривается алгоритм управления  $n$ -звенным манипуляционным роботом (МР) в среде с неизвестными статическими препятствиями. Доказывается теорема, утверждающая, что, двигаясь по данному алгоритму, МР за конечное число шагов либо достигнет целевой конфигурации, либо выдаст обоснованный ответ о том, что целевая конфигурация не может быть достигнута. Приводятся следствия из теоремы, способствующие лучшему функционированию МР в неизвестной среде.*

*Ключевые слова:* робот, неизвестная среда, препятствия, достижимость.

При управлении МР типичной является следующая задача: МР должен выдвинуться из стартовой конфигурации  $q^0$  и передвинуться в целевую конфигурацию  $q^T$ .

МР представляется в пространстве конфигураций (пространстве обобщённых координат) как точка. Функционирование МР должно происходить в пределах ограниченной области  $X$  конфигурационного пространства. Будем считать, что область  $X$  имеет такой вид, что для любого  $q \in X$  выполняются неравенства

$$a^1 \leq q \leq a^2, \quad (1)$$

где  $a^1 = (a_1^1, a_2^1, \dots, a_n^1)$  – вектор нижних ограничений на значения обобщённых координат,  $a^2 = (a_1^2, a_2^2, \dots, a_n^2)$  – вектор верхних ограничений на значения обобщённых координат,  $q = (q_1, q_2, \dots, q_n)$  – вектор обобщённых координат МР. Таким образом, область  $X$  представляет собой гиперпараллелепипед.

При этом:

1) точки, находящиеся вне (1), квалифицируются как запрещённые, и в дальнейшем мы будем называть их запрещёнными в смысле (1);

2) внутри  $X$  также могут присутствовать запрещённые состояния. Во-первых, это те состояния (конфигурации), которые обусловлены конструктивными ограничениями МР, например, те, в которых происходит недопустимое взаимопересечение звеньев. Такие запрещённые конфигурации удаётся вычислить заранее и мы будем называть их запрещёнными в смысле (2). Во-вторых, запрещённой является та конфигурация, в которой МР налегает на препятствия. В условиях неизвестной среды все такие конфигурации вычислить заранее невозможно. Перед началом движения информации о запрещённых состояниях в  $X$  нет или она неполна. Точки из  $X$ , про которые нет достоверной информации о том, что они запрещённые, считаем разрешёнными.

Рассмотрим теперь конфигурацию (точку в конфигурационном пространстве)  $q^T$ . Точку  $q^T \in X$  будем считать разрешённой только в случае, если она удовлетворяет сразу двум критериям: 1) она не является запрещённой в смыслах (1) и (2); 2) в неё можно попасть за конечное число шагов из  $q^0$ , двигаясь в  $X$  по разрешённым состояниям. Считаем, что до начала движения у МР нет достоверной информации

про точку  $q^T$  о том, является ли она разрешённой или запрещённой.

Теперь сформулируем следующую *Задачу* управления МР в неизвестной статической среде: даны стартовая конфигурация МР  $q^0 \in X$  и целевая конфигурация  $q^T \in X$ . Требуется предложить алгоритм, который за конечное число шагов либо передвинет МР из  $q^0$  в  $q^T$ , либо выдаст обоснованный ответ о том, что  $q^T$  не является разрешённой.

**Обзор литературы.** К настоящему времени имеются работы по алгоритмам управления робототехническими системами в известной и неизвестной среде. Сделаны хорошие обзоры таких алгоритмов [1–3]. Предложены алгоритмы, гарантирующие нахождение пути в среде с известными препятствиями при условии, что такой путь существует [2–5].

Некоторые алгоритмы планирования в известной среде в принципе могут быть использованы для движения в неизвестной среде. Если мы дискретизируем пространство состояний, то тогда можно будет использовать графовые методы поиска пути движения МР из  $q^0$  в  $q^T$  [2; 3; 6]. Однако эти алгоритмы имеют одно общее свойство, которое затрудняет их применение для управления МР в неизвестной среде, заключающееся в том, что они в том или ином объеме требуют осуществлять поиск в ширину, иначе не гарантируется достижение целевой точки [7]. Но при поиске в ширину часто возникает следующая ситуация: предположим, что мы только что закончили рассмотрение вершин, соседних к вершине  $q$ , и теперь нам надо рассматривать вершины, соседние вершине  $q'$ , и вершины  $q$  и  $q'$  не являются соседними. Для того чтобы рассмотреть вершины, соседние к  $q'$ , МР должен сначала передвинуться в  $q'$ . Таким образом, вновь возникает задача планирования маршрута из  $q$  в  $q'$ , что делает общую сумму перемещений МР очень большой. При планировании же в известной среде ЭВМ просто «переключает свое внимание» от  $q$  к  $q'$ , которые хранятся в памяти ЭВМ. В соответствии с классификацией [3], представителями алгоритмов поиска в ширину являются собственно алгоритм поиска в ширину, алгоритм  $A^*$ , эвристический поиск «первый-лучший», ленивый вероятностный маршрут, динамическое программирование. В методах, основанных на случайном потенциальном поле, алгоритме «Нить Ариадны», быстроисследующих случайных

деревьях [3] новые вершины генерируются случайным образом и потому применение таких методов ведет к тем же трудностям. Подходы, основанные на разделении ячеек, графах видимости [3] сводятся к попеременному построению графа и поиску пути на нем и, таким образом, им также присущ вышеописанный недостаток, связанный со множественными механическими перемещениями.

В алгоритме, представленном в данной статье, вершины  $q$  и  $q'$  всегда являются соседними, что сокращает количество движений.

Известно также, что алгоритмы поиска в глубину не всегда доводят до цели [7].

Имеется общая трудность для методов планирования пути в среде с известными препятствиями: очень трудно собрать полную информацию о рабочем пространстве робота заранее и представить эту информацию в виде, пригодном для планирования пути. При рассмотрении нашего алгоритма можно будет видеть, что для системы управления нет необходимости собирать полную информацию о рабочем пространстве заранее, МР будет собирать необходимую информацию сам в ограниченных количествах и в терминах обобщенных координат, что удобно для планирования пути.

Предпринимаются попытки создания алгоритмов управления для среды с неизвестными препятствиями. В большинстве из них рассматриваются двумерные случаи [8].

В [9–12] рассмотрены различные подходы к управлению роботами в двумерном неизвестном пространстве. В [9; 12] представлен подход, основанный на диаграмме Вороного, в [11] – поиск на основе табу-подхода. Поскольку эти подходы требуют попеременного построения графа и поиска пути на нем, они ведут к многочисленным перемещениям робота. В [10] препятствия должны иметь многоугольную форму. Применение методов, предложенных в [9–12], к управлению  $n$ -звенным манипуляционным роботом в неизвестной среде, не представлено.

В [8] представлен алгоритм управления МР среди неизвестных препятствий, расположенных в трехмерном декартовом пространстве. МР должен иметь не более трех звеньев и последняя кинематическая пара должна быть поступательной. При вышеуказанных предварительных условиях алгоритм за конечное число шагов либо переводит МР в целевую конфигурацию, либо сообщает о том, что она недостижима.

В [13] рассмотрен  $n$ -мерный случай. Алгоритм основан на решении системы нелинейных уравнений методом Ньютона и потому не может гарантировать достижения целевой позиции.

Для управления МР в неизвестной среде предлагается использовать метод искусственных потенциалов [14]. В этом методе робот представляется в виде заряженной точки, запрещенные состояния наделяются отталкивающими потенциалами, целевая точка – притягивающим потенциалом. В общем случае нет гарантии, что будет решена задача обнаружения пути, не налегающего на запрещенные точки [2].

В [3] рассмотрены алгоритмы движения роботов в присутствии неопределенности (включая случаи неизвестной среды). Алгоритмы основаны на теории последовательных решений. В общем случае алгоритмы не гарантируют достижения цели. В случаях, когда алгоритмы используют поиск на графе, возникает вышеупомянутая трудность, связанная со множественными механическими перемещениями.

**Предварительные условия.** Прежде чем привести алгоритм перемещения МР, оговорим предварительные условия.

1. Считаем, что у нас есть программная процедура  $\text{ПИ}(q^n, q^T, ZAPR)$ , где  $q^n$  есть текущая конфигурация МР, т. е. та конфигурация, в которой МР в настоящий момент находится,  $q^T$  – целевая конфигурация МР,  $ZAPR$  – множество запрещенных точек в  $X$ , сообщенных процедуре  $\text{ПИ}()$ . Может быть два исхода работы  $\text{ПИ}()$ .

1.1. Она за конечное число шагов генерирует в  $X$  путь  $L(q^n, q^T)$ , удовлетворяющий следующим условиям:

1.1.1.  $L(q^n, q^T)$  исходит из  $q^n$  и приходит в  $q^T$ ;

1.1.2.  $L(q^n, q^T) \cap ZAPR = \emptyset$ , т. е.  $L(q^n, q^T)$  ни одной своей точкой не налегает ни на одну из сообщенных запрещенных точек;

1.1.3.  $L(q^n, q^T)$  удовлетворяет (1).

В этом случае процедура  $\text{ПИ}()$  возвращает ДА.

1.2. Ей не удастся сгенерировать в  $X$  путь  $L(q^n, q^T)$ , удовлетворяющий всем трем условиям 1.1.1–1.1.3.

В этом случае процедура  $\text{ПИ}()$  возвращает НЕТ.

Для процедуры  $\text{ПИ}()$  уже существуют алгоритмы, например, алгоритм полного перебора или алгоритм  $A^*$  [6].

2. Расположение и количество препятствий внутри рабочей зоны МР остаётся неизменным в течение всего времени движения МР.

3. Всё движение, в том числе и результирующий путь, должно происходить в гиперпараллелепипеде (1).

4. МР имеет сенсорную систему (СС), которая доставляет информацию об  $r$ -окрестности текущей точки МР  $q \in X$ . Текущая точка МР – это та точка, в которой МР в настоящий момент находится. Под  $r$ -окрестностью  $q$  понимаем гипершар в  $X$  с центром в  $q$  и радиусом  $r > 0$ . Множество всех точек, входящих в  $r$ -окрестность точки  $q$ , обозначаем  $Y(q)$ . Разрешенные конфигурации из  $Y(q)$  будем помещать в множество  $Z(q)$ , запрещенные – в  $Q(q)$ .  $Z(q) \cup Q(q) = Y(q)$ . Как уже говорилось, разрешенной считается та конфигурация, которая удовлетворяет всем следующим критериям: 1) она не является запрещенной в смысле (1); 2) она не является запрещенной в смысле (2); 3) в ней ни одна точка МР не принадлежит внутренней области хотя бы одного препятствия. Если  $q$  не удовлетворяет хотя бы одному из этих трёх условий, то она считается запрещенной. Относительно каждой точки  $q^* \in Y(q)$  СС всегда доставляет достоверную информацию, является ли она разрешенной или нет. Способ записи множеств  $Y(q)$ ,  $Z(q)$ ,  $Q(q)$  может быть разным – в виде формул, списков, таб-

лиц и т. д., но мы считаем, что он есть. Устройство СС в данной работе не рассматривается.

5. Исполнение пути происходит следующим образом. МР находится в текущей точке  $q$  и, если следующая точка  $q^*$  пути удовлетворяет всем трём критериям, а именно:

- 1) она не является запрещённой в смысле (1);
- 2) не является запрещённой в смысле (2);
- 3) не налегает на препятствия,

то МР переходит в  $q^*$ . Если же в отношении хотя бы одного из критериев (1–3) получен ответ НЕТ, то в  $q^*$  МР не переходит.

**Алгоритм.** Ниже приведён алгоритм решения нашей задачи. Перед началом работы  $n = 0$ .

*Шаг 1.* МР находится в  $q^n$  (назовём её «точка смены пути»). СС доставляет информацию о  $Y(q^n)$ ,  $Z(q^n)$ ,  $Q(q^n)$ .

*Шаг 2.* Если  $q^T \in Q(q^n)$ , происходит окончание работы алгоритма с сообщением о том, что  $q^T$  является запрещённой, а значит, недостижимой. Если  $q^T \notin Q(q^n)$ , происходит переход на *Шаг 3*.

*Шаг 3.* Вызывается процедура  $\text{ПИ}(q^n, q^T, \bigcup_{i=0}^n Q(q^i))$

(т. е. нужно спланировать путь из  $q^n$  в  $q^T$ , обходящий все известные на настоящий момент запрещённые точки и лежащий целиком в (1)). Если ПИ() возвращает ДА, алгоритм переходит на *Шаг 4*, если ПИ() возвращает НЕТ, то происходит окончание работы алгоритма с сообщением о том, что  $q^T$  является недостижимой, а значит, запрещённой.

*Шаг 4.* МР начинает движение по  $L(q^n, q^T)$ . Исход движения может быть два: 1) МР попадает в разрешённую точку  $q^T$ . В этом случае происходит окончание работы алгоритма с сообщением о том, что  $q^T$  является достижимой и разрешённой; 2) МР придёт в некоторую точку  $q^*$ , следующая за которой является запрещённой. В этом случае выполняем  $n := n + 1$ ,  $q^n := q^*$  и алгоритм переходит на *Шаг 1*.

Конец алгоритма.

**Теорема.** Исполняя алгоритм, МР решит задачу за конечное число шагов.

**Доказательство.** В алгоритме, когда МР находится в точке  $q^n$ ,  $n = 0, 1, 2, \dots$ , происходит запуск СС и запуск процедуры ПИ. Если в результате исполнения этих действий точка  $q^T$  будет определена как запрещённая (в силу принадлежности множеству  $Q(q^n)$  (определяется на *Шаге 2*), либо в силу невозможности спланировать путь с помощью процедуры ПИ (определяется на *Шаге 3*)), происходит окончание работы алгоритма с сообщением о том, что  $q^T$  является запрещённой, а значит, недостижимой. Если в результате исполнения *Шага 2* и *Шага 3* точка  $q^T$  не будет определена как запрещённая, происходит генерация пути  $L(q^n, q^T)$  и МР начнёт исполнять этот путь. Исполнение этого пути может иметь два исхода: либо МР, не встретив на своём пути запрещённых точек, достигнет  $q^T$  и она окажется разрешённой (и тогда произойдёт успешное окончание работы алгоритма с со-

общением о том, что  $q^T$  является разрешённой), либо МР придёт в точку  $q^n$ ,  $n = 1, 2, \dots$ , следующая за которой будет запрещённой. Покажем, что все точки смены пути  $q^n$ ,  $n = 0, 1, 2, \dots$  будут различными и их число будет конечным.

Покажем, что все точки смены пути различны. Предположим, что МР сменил путь, находясь в точке  $q^s$ , а потом, находясь в точке  $q^p$ , вновь сменил путь, т. е.  $s < p$ . Покажем, что  $q^s \neq q^p$ . Предположим сначала, что  $q^s = q^p$  и тогда  $Q(q^s) = Q(q^p)$ . Поскольку МР сменил путь, находясь в точке  $q^s$ , то он сгенерировал путь, не налегающий в том числе и на  $Q(q^s)$ . Но поскольку МР сменил путь в точке  $q^p$ , то это означает, что его путь налёг на  $Q(q^p) = Q(q^s)$  (при этом  $q^s = q^p$  является центром  $r$ -окрестности точки  $q^s = q^p$  и следующая за ней точка является запрещённой), т. е.  $Q(q^p) = Q(q^s)$  не было известным. Получили противоречие. Отсюда видно, что все точки смены пути различны.

Покажем теперь, что число точек смены пути конечно. Предположим, что, наоборот, оно бесконечно. Все точки смены пути должны удовлетворять ограничениям (1). Это означает, что последовательность этих точек ограничена. Согласно теореме Больцано–Вейерштрасса, из этой последовательности можно извлечь сходящуюся подпоследовательность  $q^i$ ,  $i = 1, 2, \dots$  В соответствии со свойством Коши сходящихся последовательностей, для любого  $\varepsilon$  можно найти такой номер  $s$ , что все точки  $q^i$ ,  $i > s$ , будут лежать в  $\varepsilon$ -окрестности точки  $q^s$ . Возьмем  $\varepsilon < r$ . Рассмотрим произвольную точку смены пути  $q^i$ , расположенную в  $\varepsilon$ -окрестности точки  $q^s$ . Так как МР, находясь в  $q^i$ , сменил свой путь, то это означает, что путь налёг на множество  $Q(q^s)$  ( $q^i$  и ее соседние точки принадлежат  $Q(q^s)$ ). Отсюда надо сделать вывод, что множество  $Q(q^s)$  не было учтено при генерации пути, приведшего в  $q^i$ , что невозможно при исполнении предписаний алгоритма. Таким образом, если принять, что число точек смены пути бесконечно, то неизбежно возникнет ситуация, которая не может наступить при исполнении предписаний алгоритма. Следовательно, число точек смены маршрута конечно.

Итак, число точек смены пути  $q^n$ ,  $n = 0, 1, 2, \dots$  конечно, и они все различны. В каждой точке  $q^n$  осуществляется запуск СС и вызов процедуры ПИ(), генерирующей  $L(q^n, q^T)$ . В результате выполнения этих действий либо получаем информацию о том, что  $q^T$  является запрещённой, либо нет. Если получаем, выяснение вопроса о достижимости  $q^T$  заканчивается выводом о недостижимости  $q^T$ . Если не получаем, происходит планирование пути  $L(q^n, q^T)$  и его исполнение. Если в последней точке смены пути  $q^n$  путь в  $q^T$  был спланирован, этот путь будет исполнен,  $q^T$  будет достигнута и она окажется разрешённой.

В каждой точке смены пути  $q^n$  происходит вызов процедуры ПИ() для планирования пути  $L(q^n, q^T)$  с последующим его исполнением. Число вызовов процедуры ПИ() конечно. Планирование пути на любом

вызове ПИ() происходит за конечное число шагов. Исполнение любого пути, спланированного в ПИ(), происходит за конечное число шагов в силу конечности длины пути. Отсюда видно, что решение задачи сводится к решению конечного числа задач планирования и исполнения пути в среде с известными запрещёнными состояниями и осуществится за конечное число шагов.

Теорема доказана.

Иногда требуется интенсивное исследование окружающей среды, предполагающее сход с предварительно спланированного пути. Для таких случаев будут полезны приводимые ниже следствия.

**Следствие 1.** В произвольных точках  $q^d, d=0, 1, \dots, N_{dn}$  исполняемого пути  $L(q^n, q^T)$ , где  $n=0, 1, \dots, N_{dn}$  – некоторое конечное число, не большее, чем число точек в  $L(q^n, q^T)$ , МР может сделать конечное число сходов с  $L(q^n, q^T)$ . Сходом является последовательность  $S$  непрерывно следующих одна за другой разрешённых точек, отличающаяся от  $L(q^n, q^T)$  и удовлетворяющая (1). Сходы делаются для исследования окружающей среды. Число шагов на каждом сходе должно быть конечным. На каждом шаге схода МР может исследовать окружающую среду. Осуществив конечное число шагов на сходе, МР должен вернуться на первоначальный путь точно по ранее пройденным точкам схода. Если на каком-то шаге схода обнаруживается, что дальнейшее движение невозможно в силу наличия запрещённых состояний, МР должен вернуться на первоначальный путь точно по ранее пройденным точкам схода. Таким образом, смены пути не происходит. Если МР вернётся не в точку  $q^d \in L(q^n, q^T)$ , с которой он сошёл, а на какую-то другую точку  $q^e \in L(q^n, q^T)$ , то точки из  $L(q^n, q^T)$  между  $q^d$  и  $q^e$  могут оказаться запрещёнными и, когда впоследствии МР попытается исполнить  $L(q^n, q^T)$ , он натолкнётся на эти ранее не обнаруженные запрещённые точки. По возвращении на  $L(q^n, q^T)$  МР осуществляет движение в соответствии с алгоритмом. В результате мы получаем конечное число шагов на всех сходах, которое прибавляется к конечному числу шагов по алгоритму.

**Следствие 2.** МР также может, находясь в какой-либо точке  $q$  пути или схода, получив информацию о  $Q(q)$ , предварительно выполнив  $n: = n + 1$  и  $q^n = q$  спланировать новый путь  $L(q^n, q^T)$  путём вызова процедуры ПИ( $q^n, q^T, \bigcup_{i=0}^n Q(q^i)$ ) и начать двигаться по нему в соответствии с алгоритмом. Иными словами, это такое действие (назовем его «отказ от предварительного пути»), когда, не достигнув  $q^T$ , планируют новый путь, ведущий в  $q^T$ . Делать отказы от предварительного пути можно лишь конечное число раз, а после этого МР, пройдя конечное число шагов по разрешённым точкам и находясь в некотором состоянии  $q$ ,

должен выполнить  $n: = n + 1$  и  $q^n = q$  и перейти к выполнению алгоритма, гарантирующего решение задачи за конечное число шагов. Тогда конечное число отказов от предварительного пути прибавится к конечному числу шагов по алгоритму, гарантирующему решение задачи за конечное число шагов, и в результате мы получим, что задача будет решена за конечное число шагов.

### Библиографические ссылки

1. Ahrikhencheikh C., Seireg A. Optimized-Motion Planning: Theory And Implementation. Chichester : John Wiley & Sons, 1994.
2. Principles of Robot Motion. Theory, Algorithms and Implementations / H. Choset [et al.]. Cambridge, Mass. : MIT Press, 2005.
3. LaValle S. M. Planning Algorithms. 1999–2006. URL: <http://msl.uiuc.edu/planning> (data of visit: 30.06.2011).
4. Canny J. The Complexity of Robot Motion Planning. Cambridge, Mass. : MIT Press, 1988.
5. Collins G. E. Quantifier Elimination For Real Closed Fields By Cylindrical Algebraic Decomposition // Lecture Notes in Computer Sci. 1975. Vol. 33. P. 135–183.
6. Нильсон Н. Искусственный интеллект. М. : Мир, 1973.
7. Ильин В. А. Интеллектуальные роботы: теория и алгоритмы / Сиб. аэрокосмич. акад. – Красноярск, 1995.
8. Lumelsky V. J. Sensing, intelligence, motion. How robots and humans move in an unstructured world. N. Y. : John Wiley & Sons, 2006.
9. Chen C., Li H.-X., Dong D. Hybrid Control for Robot Navigation. A hierarchical Q-learning algorithm // IEEE Robotics & Automation Magazine. 2008. Vol. 15. № 2. P. 37–47.
10. Online Algorithms with Discrete Visibility. Exploring Unknown Polygonal Environments / S. K. Ghosh, J. W. Burdick, A. Bhattacharya, S. Sarkar // IEEE Robotics & Automatiom Magazine. 2008. Vol. 15. № 2. P. 67–76.
11. Masehian E., Amin-Nasari M. R. Sensor-Based Robot Motion Planning. A Tabu Search Approach // IEEE Robotics & Automation Magazine. 2008. Vol. 15. № 2. P. 48–57.
12. Rawlinson D., Jarvis R. Ways to Tell Robots Where to Go. Directing autonomous robots using topological instructions // IEEE Robotics & Automation Magazine. 2008. Vol. 15. № 2. P. 27–36.
13. Yegenoglu F., Erkmen A. M., Stephanou H. E. On-line Path Planning Under Uncertainty // Proc. of 27th IEEE Conf. Decis. and Contr. Austin, Tex. Dec. 7–9. 1988. Vol. 2. P. 1075–1079.
14. Barraquand J., Latombe J.-C. Robot Motion Planning: A Distributed Representation Approach // Int. J. of Rob. Res. 1991. Vol. 10. № 6. P. 628–649.

P. K. Lopatin

## ALGORITHM OF A MANIPULATING ROBOT MOVEMENT IN AN ENVIRONMENT WITH UNKNOWN OBSTACLES

An algorithm is presented for a manipulating robot (MR) control in an unknown static environment. A theorem is proved stating that following the algorithm the MR in a finite number of steps will either reach a target configuration or come to the proved conclusion that the target configuration may not be reached. Given sequences from the theorem facilitating the MR functioning in the unknown environment.

Keywords: robot, unknown environment, obstacles, reachability.

© Лопатин П. К., 2011

УДК 62.501

А. В. Медведев

## ТЕОРИЯ НЕПАРАМЕТРИЧЕСКИХ СИСТЕМ. К-МОДЕЛИ

Рассматриваются некоторые задачи идентификации в «широком» смысле. Идентификация безынерционных объектов и объектов с памятью исследуется в условиях непараметрической априорной информации. Анализируется также случай, когда априорная информация соответствует как непараметрическому, так и параметрическому уровню одновременно. Это относится к системе многосвязных объектов. Приводятся непараметрические модели некоторых статических и динамических объектов с запаздыванием.

Ключевые слова: идентификация, априорная информация, идентификация в «широком», непараметрические модели, дискретно-непрерывные процессы, К-модели.

Теория – в виду практики.  
Девиз конгрессов IFAC

При изучении неизвестных явлений можно строить очень общие гипотезы и продвигаться шаг за шагом с помощью опыта. Слишком конкретная гипотеза почти наверняка заключает в себе долю ошибки наряду с долей истины

П. Кюри

Проблема идентификации является одной из центральных в теории управления и других областях науки, объединяемых емким термином – кибернетика. Основное внимание мы уделим задачам идентификации в «широком» смысле, наряду с достаточно хорошо развитой теорией идентификации в «узком» смысле. Ранее [1–3] были описаны дискретно-непрерывные процессы и пути идентификации стохастических систем, которые тесно связаны с имеющейся априорной информацией.

Более того, нас будет интересовать прежде всего идентификация в условиях непараметрической неопределенности, а также случай, когда априорная информация соответствует одновременно как непараметрическому [3], так и параметрическому типу исходных данных об исследуемом процессе. Приведем достаточно общепринятую схему задачи идентификации (рис. 1) [4].

На рисунке приняты следующие обозначения:  $u_t$ ,  $x_t$  – соответственно векторные входные и выходные переменные, контролируемые в дискретные моменты времени  $t$ ;  $\xi_t$  – случайная ненаблюдаемая помеха;

$\hat{x}_t$  – выходная величина настраиваемой модели;  $\varepsilon_t = x_t - \hat{x}_t$ ;  $Q(\varepsilon_t)$  – выпуклая функция потерь;  $\alpha$  – вектор параметров настраиваемой модели;  $I_t$  – вектор всех наблюдений к моменту времени  $t$ ;  $M$  – символ математического ожидания;  $R(\alpha)$  – критерий идентификации.

Идентификация по данной схеме осуществляется при помощи настраиваемой модели (блок «модель») той или иной параметрической структуры, параметры  $\alpha$  которой корректируются по мере поступления наблюдений «вход-выход». Обычно предполагается, что объект работает в стационарном режиме, т. е. вероятностные характеристики последовательностей  $\bar{x}_s$ ,  $\bar{u}_s$  неизменны во времени,  $\bar{x}_s = (x_1, x_2, \dots, x_s)$ ,  $\bar{u}_s = (u_1, u_2, \dots, u_s)$  – временные векторы. Соответствие настраиваемой модели объекту оценивается критерием качества идентификации:

$$R(\alpha) = M \{Q(\varepsilon(x_t, \hat{x}_t))\}. \quad (1)$$