АЛГОРИТМ ЗАХВАТА МАНИПУЛЯТОРОМ ОБЪЕКТА В НЕИЗВЕСТНОЙ СТАТИЧЕСКОЙ СРЕДЕ

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 захватит объект, либо выдаст обоснованный ответ о том, что объект не может быть захвачен ни в одной из конфигураций.

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

При управлении МР типичной является следующая задача: МР должен выдвинуться из стартовой конфигурации q^0 и захватить своим схватом некоторый объект. При этом часто данный объект может быть захвачен не в одной, а в нескольких, а иногда и в бесконечном количестве целевых конфигураций q_i^T . Целевые конфигурации q_i^T объединяются в целевое множество B_T . Множество B_T имеет произвольный вид.

Будем считать, что B_T не пополняется в течение всего времени движения МР. Считаем также, что координаты каждой точки из B_T известны и определены достоверно.

МР представляется в пространстве конфигураций (пространстве обобщенных 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 обобщенных координат; $a^2 = (a_1^2, a_2^2, \dots, a_n^2)$ – вектор верхних ограничений на значения обобщенных координат; $q = (q_1, q_2, \dots, q_n)$ – вектор обобщенных координат МР. Таким образом, область X представляет собой гиперпараллелепипед. Все точки, не удовлетворяющие (1), будем считать запрещенными.

Кроме того, следует учитывать, что и внутри X могут присутствовать запрещенные состояния. Во-первых, это те состояния (конфигурации), которые обусловлены конструктивными ограничениями МР, например те, в которых происходит недопустимое взаимопересечение звеньев. Такие запрещенные конфигурации удается вычислить заранее. Во-вторых, запрещенной является та конфигурация, в которой МР налегает на препятствия. В условиях неизвестной среды все такие конфигурации вычислить заранее невозможно. Итак, запрещенной конфигурацией является та конфигурация МР, в которой МР не может находиться в силу конструктивных ограничений либо в силу налегания на препятствие. Перед началом движения информации о запрещенных состояниях в X нет или она неполна. Точки из X , про которые нет достоверной информации о том, что они запрещенные, считаем разрешенными.

Рассмотрим теперь точки из B_T . Разрешенной точкой $q^T \in B_T$ будем считать только ту точку, которая удовлетворяет следующим двум критериям: 1) она не является запрещенной в смыслах, описанных в предыдущих абзацах;

2) в нее можно попасть за конечное число шагов из q^0 , двигаясь в X по разрешенным состояниям. Точки из B_T , не удовлетворяющие хотя бы одному из двух таких критериев, считаем запрещенными. Поскольку требуется выяснить, достижимо ли множество B_T хотя бы в одной точке, то считаем, что до начала движения у МР ни про одну точку из B_T нет достоверной информации о том, является ли она разрешенной или запрещенной. Теперь сформулируем следующую **Задачу** управления МР в неизвестной статической среде: даны стартовая конфигурация МР q^0 и целевое множество B_T . Требуется предложить алгоритм, который за конечное число шагов либо передвинет МР из q^0 в хотя бы одно состояние из множества B_T , либо выдаст обоснованный ответ о том, что ни одно состояние из целевого множества B_T не является разрешенным.

К настоящему времени имеются работы по алгоритмам управления динамическими (ДС) и, в частности, робототехническими (РС) системами в известной и неизвестной среде. Имеются хорошие обзоры таких алгоритмов [1; 2]. Предложены алгоритмы, например, алгоритм полного перебора и алгоритм A^* , которые за конечное число шагов либо находят путь из q^0 в точку из B_T , либо сообщают, что в B_T нет точки, достижимой из q^0 .

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 траектории движения ДС [2; 3]. Однако эти алгоритмы имеют одно общее свойство, которое затрудняет их применение для управления ДС в неизвестной среде, заключающееся в том, что они в том или ином объеме требуют осуществлять поиск в ширину, иначе не гарантируется достижение целевой точки [4]. 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 к вершине q' и вершины q и q' не являются соседними. Для того, чтобы рассмотреть вершины, соседние к q' , ДС должна сначала передвинуться в q' . Таким образом, вновь возникает задача планирования и осуществления маршрута из q в q' . При планировании же в известной среде ЭВМ просто «переключает свое внимание» от q к q' , которые хранятся в памяти ЭВМ. Необходимость поиска и исполнения путей для многих различных q и q' делает общую сумму передвижений ДС очень большой [4]. В случае, когда мы

планируем путь в известной среде, компьютер просто переключает свое внимание от точки q к точке q' , которые хранятся в памяти компьютера.

В соответствии с классификацией [2] 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», быстро исследующих случайных деревьях [2] новые вершины генерируются случайным образом и потому применение таких методов ведет к тем же трудностям. Подходы, основанные на разделении ячеек, (двунаправленных) графах видимости, диаграммах Вороного [2] сводятся к попеременному построению графа и поиску пути на нем, таким образом, им также присущ вышеописанный недостаток, связанный со множественными механическими перемещениями.

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

Известно также, что алгоритмы поиска в глубину не всегда доводят до цели [4].

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 разработки создания алгоритмов управления для среды с неизвестными препятствиями. Большинство из них рассматривают двумерные случаи [5].

В [6–9] рассмотрены различные подходы к управлению роботами в двумерном неизвестном пространстве. В [6; 9] представлен подход, основанный на диаграмме Вороного, в [8] – поиск на основе табу-подхода. Поскольку эти подходы требуют попеременного построения графа и поиска пути на нем, они ведут к многочисленным перемещениям робота. В [7] препятствия должны иметь многоугольную форму. Применение методов, предложенных в [6–9] к управлению n -звенным манипуляционным роботом в неизвестной среде, не представлено.

В [5] 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 [10] рассмотрен n -мерный случай. Алгоритм основан на решении системы нелинейных уравнений методом Ньютона и потому не может гарантировать достижения целевой позиции.

В [2] 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.

В [11] предлагается алгоритм управления ДС в n -мерном пространстве состояний в присутствии неизвестных запрещенных статических состояний, за конечное число шагов либо переводящий ДС из стартовой точки q^0 в целевую q^T , либо выдающий обоснованное сообщение о том, что q^T недостижима. Алгоритм обладает таким недостатком: он предполагает, что во множестве X каждое состояние достижимо из каждого. Данное требование в ряде случаев может входить в противоречие с целью самого алгоритма – за конечное число шагов выяснить достижимость или недостижимость целевого состояния из стартового.

В [4] предложен подход к управлению роботами (в том числе и манипуляционными) в n -мерном пространстве состояний. Суть подхода заключается в том, что робот планирует путь, соединяющий стартовую точку q^0 и целевую q^T , обходящий известные запрещенные состояния, и пытается исполнить данный путь либо до достижения q^T , либо до столкновения в некоторой точке q^n с ранее неизвестным запрещенным состоянием. В последнем случае планируется новый путь $L(q^n, q^T)$, соединяющий q^n и q^T и обходящий все известные запрещенные состояния. Показано [4], что задача перевода робота из q^0 в q^T в неизвестной среде сводится к решению конечного числа задач планирования и исполнения пути $L(q^n, q^T)$ (другими словами, к решению конечного числа задач ПИ (планирования в известной среде)). При этом предполагается, что имеется априорное знание о том, что q^T достижима.

В настоящей статье при решении **Задачи** мы опирались на подход [4]. Сформулированная **Задача** нами сводится к задаче исследования достижимости конечного числа N_{BT} точек $q_i^T, i = 1, 2, \dots, N_{BT}$. Требование априорного знания о достижимости $q_i^T, i = 1, 2, \dots, N_{BT}$ снято. Сформулированы новые требования к процедуре ПИ, необходимые для решения **Задачи**.

В [12–15] даются алгоритмы, за конечное число шагов переводящие ДС в среде с неизвестными запрещенными статическими состояниями из q^0 в q^T . Однако при этом предполагалось, что имеется априорная информация о том, что q^T достижима.

Рассмотрим предварительные условия:

1. На множестве B_T выделяем конечное число N_{BT} точек $q_i^T, i = 1, \dots, N_{BT}$. Это те целевые конфигурации, достижимость которых будет исследоваться. В дальнейшем B_T будем рассматривать как список конфигураций $q_i^T, i = 1, \dots, N_{BT}$. Считаем, что B_T не пополняется и потому N_{BT} не увеличивается. Считаем, что координаты каждой точки из B_T определяются достоверно.

2. Считаем, что у нас есть процедура ПИ, которая за конечное число шагов либо планирует траекторию из произвольной разрешенной точки $q^n \in X$ в произвольную точку $q^T \in X$ среди сообщенных ей известных запрещен-

ных состояний, либо выдает сообщение, что q^T недостижима. Такие процедуры уже существуют, например алгоритм полного перебора или алгоритм A^* [3], которые для любой начальной точки q^n и любой целевой q^T при заданных известных запрещенных состояниях за конечное число шагов либо находят путь из q^n в q^T , либо сообщают, что путь из q^n в q^T найден быть не может.

3. Расположение препятствий внутри рабочей зоны МР остается неизменным в течение всего времени движения МР.

4. Количество препятствий внутри рабочей зоны МР остается неизменным в течение всего времени движения МР.

5. Все движение, в том числе и результирующая траектория, должно происходить в гиперпараллелепипеде (1).

6. 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 $Y(q)$. Слова «доставляет информацию об r -окрестности точки q » означают, что относительно каждой точки из $Y(q)$ СС определяет, является ли она разрешенной или запрещенной, при этом все запрещенные точки из $Y(q)$ заносятся в множество $Q(q)$, а все разрешенные точки из $Y(q)$ заносятся в множество $Z(q)$. Способ записи множеств $Y(q)$, $Z(q)$, $Q(q)$ может быть разным – в виде формул, списков, таблиц и т. д., но мы считаем, что он есть. Устройство сенсорной системы в данной работе не рассматривается.

7. Считаем, что у нас есть программная Процедура1($B_T, N_{BT}, Q(q^n)$). Процедура1($B_T, N_{BT}, Q(q^n)$) получает при вызове множество B_T , количество N_{BT} точек в множестве B_T , множество $Q(q^n)$, полученное при последнем запуске СС. Процедура1($B_T, N_{BT}, Q(q^n)$) выбрасывает из B_T те точки, которые совпадают с точками из $Q(q^n)$. После выброса оставшиеся точки в B_T перенумеровываются сплошной нумерацией начиная с 1 и в N_{BT} записывается число точек, оставшихся в B_T после исполнения Процедуры1($B_T, N_{BT}, Q(q^n)$).

8. Считаем, что у нас есть программная Процедура2(B_T, N_{BT}, q^T). Процедура2(B_T, N_{BT}, q^T) получает при вызове множество B_T , количество N_{BT} точек в множестве B_T и точку q^T . Процедура2(B_T, N_{BT}, q^T) выбрасывает из B_T точку q^T . После этого точки перенумеровываются сплошной нумерацией начиная с 1 и в N_{BT} записывается число точек, оставшихся в B_T после исполнения Процедуры2(B_T, N_{BT}, q^T).

Ниже приведен Алгоритм решения нашей Задачи. Перед началом движения текущей конфигурацией q^c МР является q^0 , по ходу движения Алгоритм1 может вызываться из других текущих конфигураций МР.

Алгоритм. Если $N_{BT}=0$, то происходит окончание работы алгоритма с сообщением о том, что объект захвачен быть не может. Иначе в качестве q^T назначаем первую точку из B_T .

Шаг 1. МР находится в конфигурации q^c . $n:=0$, $q^n:=q^c$. Исполнить

целевая_точка_запрещенная:=Алгоритм1(q^c, q^T, B_T, N_{BT}).

Если целевая_точка_запрещенная=НЕТ, происходит успешное окончание работы Алгоритма с сообщением о том, что объект захвачен в точке q^T .

Если целевая_точка_запрещенная=ДА, происходит переход на шаг 2.

Шаг 2. Если $N_{BT}=0$, происходит окончание работы алгоритма с сообщением о том, что объект не может быть захвачен МР ни в одной из целевых конфигураций. Если $N_{BT} \neq 0$, в качестве q^T взять первую точку из B_T и перейти на шаг 1. **Конец Алгоритма.**

Алгоритм1 получает значения в следующем формате: Алгоритм1(q^n, q^T, B_T, N_{BT}) и посвящен выяснению вопроса о том, является ли точка q^T достижимой из q^n в известной среде.

Алгоритм1. Шаг 1. МР находится в q^n (назовем ее «точка смены маршрута»). СС доставляет информацию о $Y(q^n)$, $Z(q^n)$, $Q(q^n)$.

Шаг 2. Выполняется

$N_{BT} := \text{Процедура1}(B_T, N_{BT}, Q(q^n))$.

Если $N_{BT}=0$, происходит присвоение целевая_точка_запрещенная:=ДА и возврат в алгоритм. Если $N_{BT} \neq 0$, проверяем, оказалась ли выброшенной из B_T точка q^T . Если да, то происходит присвоение целевая_точка_запрещенная:=ДА и возврат в алгоритм, если нет (т. е. q^T осталась в B_T) – то происходит переход на шаг 3.

Шаг 3. Вызывается процедура ПИ с задачей сгенерировать линию $L(q^n, q^T)$, удовлетворяющую следующим условиям:

– $L(q^n, q^T)$ соединяет q^n и q^T ;

– $L(q^n, q^T)$ не налегает на множество $\bigcup_{i=0}^n Q(q^i)$, т. е. ни на одну из запрещенных точек;

– $L(q^n, q^T)$ удовлетворяет конструктивным ограничениям (1).

Результатов работы процедуры ПИ может быть два:

– ПИ возвращает сгенерированную траекторию и Алгоритм1 переходит на шаг 4;

– ПИ сообщает о том, что $L(q^n, q^T)$ сгенерирована быть не может, т. е. q^T является недостижимой. В этом случае выполняется

$N_{BT} := \text{Процедура2}(B_T, N_{BT}, q^T)$,

производится присвоение: целевая_точка_запрещенная:=ДА и происходит возврат в Алгоритм1.

Шаг 4. МР начинает движение по $L(q^n, q^T)$. Исходов движения может быть два:

– МР попадает в некоторую точку $q_i^T \in B_T$. В этом случае происходят присвоения $q^T:=q_i^T$ и целевая_точка_запрещенная:=НЕТ и выполняется возврат в Алгоритм1;

– МР придет в некоторую точку q^* , следующая за которой является запрещенной. В этом случае выполняем $n:=n+1$, $q^n:=q^*$ и Алгоритм1 переходит на шаг 1. **Конец Алгоритма1.**

Теорема. Исполняя Алгоритм, МР решит Задачу за конечное число шагов.

Доказательство. Алгоритм выясняет достижимость конечного числа точек из B_T . Выяснение достижимости в отношении каждой из точек $q^T \in B_T$ осуществляется путем исполнения Алгоритма1. Отсюда видно, что исполнение Алгоритма сводится к конечному числу вызовов Алгоритма1. Поэтому, чтобы доказать, что Алгоритм будет исполнен за конечное число шагов, тре-

буется показать, что исполнение *Алгоритма 1* для произвольных q^n и q^T будет осуществлено за конечное число шагов.

Алгоритм 1 посвящен выяснению вопроса о том, является ли точка q^T достижимой в неизвестной среде из точки смены маршрута q^n или нет. В *Алгоритме 1*, 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 запрещенная (в силу налегания на препятствие, либо в силу недостижимости), произойдет возврат в алгоритм и для исследования достижимости будет назначена другая точка $q^T \in B_r$. Если в результате исполнения этих действий точка q^T не будет определена как запрещенная, происходит генерация линии $L(q^n, q^T)$ и МР начнет исполнять эту траекторию. Исполнение этой траектории может иметь два исхода: либо МР, не встретив на своем пути запрещенных точек, достигнет q^T (и тогда произойдет успешное окончание работы *Алгоритма*) либо МР придет в точку q^n , $n = 1, 2, \dots$, следующая за которой будет запрещенной. Покажем, что все точки смены маршрута q^n , $n = 0, 1, 2, \dots$ будут различными и их число будет конечным.

Покажем, что все точки смены маршрута различны. Предположим, что МР сменил маршрут, находясь в точке q^s , а потом, находясь в точке q^p , вновь сменил маршрут, 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 маршрут, находясь в точке q^s , то он сгенерировал маршрут, не налегающий в том числе и на $Q(q^s)$. Но поскольку МР сменил маршрут в точке q^p , то это означает, что его маршрут налег 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 запрещенной), т. е. $Q(q^p) = Q(q^s)$ не было известным. Получили противоречие. Отсюда видно, что все точки смены маршрута различны.

Покажем теперь, что число точек смены маршрута конечно. Предположим, что, наоборот, оно бесконечно. Все точки смены маршрута 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 ϵ можно найти такой номер s , что все точки q^i , $i > s$, будут лежать в ϵ -окрестности точки q^s . Возьмем $\epsilon < r$. Рассмотрим произвольную точку смены маршрута q^i , расположенную в ϵ -окрестности точки q^s . Так как МР, находясь в q^i , сменил свой маршрут, то это означает, что маршрут налег 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 маршрута, приведшего в q^i , что невозможно при исполнении предписаний *Алгоритма 1*. Таким образом, если принять, что число точек смены маршрута бесконечно, то неизбежно возникнет ситуация, которая не может наступить при исполнении предписаний *Алгоритма 1*. Следовательно, число точек смены маршрута конечно.

Итак, число точек смены маршрута q^n , $n = 0, 1, 2, \dots$ конечно и они все различны. В каждой точке q^n осуществляется запуск СС и вызов процедуры ПИ, генерирую-

щей $L(q^n, q^T)$. В результате выполнения этих действий либо получаем информацию о том, что q^T является запрещенной, либо нет. Если получаем, выяснение вопроса о достижимости q^T заканчивается выводом о недостижимости q^T . Если нет, происходит попытка исполнения траектории $L(q^n, q^T)$. Если и в последней точке смены маршрута q^n точка q^T не была квалифицирована как запрещенная, то будет сгенерирована $L(q^n, q^T)$, эта траектория будет исполнена и q^T будет достигнута.

Таким образом, показано, что МР, исполняя *Алгоритм 1* за конечное число шагов либо достигнет точки q^T , либо сделает вывод о том, что q^T недостижима. *Алгоритм* сводится к исполнению *Алгоритма 1* конечное число раз. Отсюда видно, что, исполняя *Алгоритм*, МР решит задачу за конечное число шагов. *Теорема доказана.*

Замечание 1. Мы уже говорили, что *Алгоритм 1* сводится к генерации и исполнению конечного числа пути $L(q^n, q^T)$. Алгоритм сводится к исполнению конечного числа раз *Алгоритма 1*. Отсюда следует вывод о том, что **Задача** сводится к решению конечного числа задач ПИ планирования и исполнения пути в среде с известными запрещенными состояниями.

Замечание 2. При первом вызове *Алгоритма 1* $q^c = q^0$ исследуется достижимость первой точки из B_r , назовем ее q_1^T . При последующих вызовах *Алгоритма 1* исследуется достижимость точек $q_i^T \in B_r$, $i = 2, 3, \dots, N_{BT}$ и, вообще говоря, $q^c \neq q^0$. Но, поскольку МР прибыл в q^c из q^0 по непрерывно следующим одна за другой разрешенным точкам, то вывод о достижимости/недостижимости $q_i^T \in B_r$, $i = 2, 3, \dots, N_{BT}$ из $q^c \neq q^0$ будет квалифицироваться как вывод о достижимости/недостижимости точки $q_i^T \in B_r$, $i = 2, 3, \dots, N_{BT}$ из q^0 .

Библиографические ссылки

1. Ahrikhencheikh C., Seireg A. Optimized-Motion Planning: Theory And Implementation. John Wiley & Sons, Inc, 1994.
2. LaValle S. M. Planning Algorithms. 1999–2006. [Electronic resource]. URL: <http://msl.cs.uiuc.edu/planning>.
3. Ильин В. А. Интеллектуальные роботы: Теория и алгоритмы / Сиб. аэрокосмич. акад. Красноярск, 1995.
3. Nilson N. Problem-Solving Methods in Artificial Intelligence // New York : McGraw-Hill Book Company, 1971.
5. Lumelsky V. J. Sensing, intelligence, motion. How robots and humans move in an unstructured world / John Wiley & Sons, 2006.
6. Chen C., Li H.-X., Dong D. Hybrid Control for Robot Navigation. A hierarchical Q-learning algorithm // IEEE Robotics & Automation Magazine. 2008. June. Vol. 15, № 2. P. 37–47.
7. Online Algorithms with Discrete Visibility. Exploring Unknown Polygonal Environments / S. K. Ghosh [et al.] // IEEE Robotics & Automiom Magazine. 2008. June. Vol. 15, № 2. P. 67–76.
8. Masehian E., Amin-Nasari M. R. Sensor-Based Robot Motion Planning. A Tabu Search Approach // IEEE Robotics & Automation Magazine. 2008. June. Vol. 15, № 2. P. 48–57.
9. Rawlinson D., Jarvis R. Ways to Tell Robots Where to Go. Directing autonomous robots using topological

instructions // IEEE Robotics & Automation Magazine. 2008. June. Vol. 15, № 2. P. 27–36.

10. Yegenoglu F., Erkmen A. M., Stephanou H. E. On-line Path Planning Under Uncertainty. Proc. 27th IEEE Conf. Decis. and Contr. (Austin, Tex., Dec. 7–9, 1988). Vol. 2. New York, 1988. P. 1075–1079.

11. Отчет о госбюджетной научно-исследовательской работе Б6-12 за 1996 г. Алгоритмическое и программное обеспечение интеллектуальных манипуляционных роботов в условиях неопределенности. Теория и моделирование задач управления, планирования траекторий и действий манипуляционными интеллектуальными роботами в условиях неопределенности : отчет о НИР / Сиб. аэрокосмич. акад. ; Ильин В. А. Красноярск, 1997. № ГР 01.9.80.001231. Инв. № 02.9.8.0 000526.

12. Лопатин П. К. Компьютерная имитация управления манипуляционными роботами в неизвестной среде на основе точного и упрощенного алгоритмов // Мехатроника, автоматизация, управление. Приложение. 2006. № 8. С. 7–14.

13. Лопатин П. К. Алгоритм2 управления динамическими системами в неизвестной статической среде // Вестник СибГАУ. Вып. 4 (11). 2006. С. 28–32.

14. Лопатин П. К. Алгоритм управления динамическими системами в неизвестной статической среде // Мехатроника, автоматизация, управление. 2007. № 2. С. 9–13.

15. Lopatin P. K. Algorithm of a manipulator movement amidst unknown obstacles // Proc. of the 10 th Intern. Conf. on Advanced Robotics (ICAR 2001) (Budapest, Aug. 22–25, 2001). Hungary, 2001. P. 327–331.

P. K. Lopatin

ALGORITHM FOR AN OBJECT GRASPING BY A MANIPULATOR IN AN UNKNOWN STATIC ENVIRONMENT

An algorithm for n-link manipulating robot (MR) control in environment with unknown static obstacles is considered and theorem, which states that following the algorithm, MR for finite number of steps either grasps the object or gives proved conclusion that the object cannot be grasped in no configuration, is proved in the article.

Keywords: robot, unknown environment, obstacles, reachability.

© Лопатин П. К., 2010

УДК 004.4'2

А. Ю. Сыщиков

КОНЦЕПЦИЯ И ТЕХНОЛОГИЯ ПАРАЛЛЕЛЬНОГО ПРОГРАММИРОВАНИЯ НЕОДНОРОДНЫХ СИСТЕМ НА КРИСТАЛЛЕ

Неоднородные вычислительные системы на кристалле – это быстрорастущий сегмент распределенных параллельных вычислительных систем. Такие системы не могут эффективно программироваться вручную, тем не менее существуют только базовые инструменты для их программирования. Предлагается концепция и технология параллельного программирования неоднородных систем на кристалле. Концепция представляет визуальный подход к программированию, среднегранулярную организацию параллельных вычислений с последовательными блоками и динамиче-ску параллельных вычислений. Представлен разработанный пакет инструментов VIPE (VPL Integrated Programming Environment). VIPE включает в себя формальную математическую модель параллельных вычислений, визуальный язык параллельного программирования для среднегранулярного дизайна программ, транслятор и прекомпилятор схем программ и программный симулятор выполнения параллельных программ на неоднородной системы на кристалле.

Ключевые слова: система на кристалле, параллельное программирование, динамические вычисления, интегрированная среда разработки.

Многопроцессорные системы на кристалле (Multiprocessor Systems-on-Chips, СнК) – одно из ключевых современных приложений технологий сверхбольших интегральных схем. Требования приложений и ограничения реализации вынуждают создавать заказные, нестандартные, неоднородные архитектуры для оптимизации скорости вычислений, энергопотребления, стоимости, размеров и других критериев.

Одним из важнейших вопросов в области разработки СнК является технологии и средства программирования таких систем. В коллективах, создающих СнК, существуют программные средства, позволяющие разработать необходимый код для функционирования системы: компиляторы для ядер, загрузчики, возможно, отладчики. Но такой подход является фрагментарным: части системы программируются независимо, взаимосвязи между ними