

АЛГОРИТМ РАЗДЕЛЕНИЯ ЯЧЕЕК ДЛЯ ПЛАНИРОВАНИЯ ПУТИ МАНИПУЛЯЦИОННОГО РОБОТА

Используется известный подход к управлению манипуляционными роботами в неизвестной среде, который сводится к решению конечного числа задач планирования пути в известной среде. Для планирования пути в среде с известными запрещенными состояниями применяется алгоритм деления ячеек, пригодный для n -мерного пространства обобщенных координат. Представлены результаты программной реализации алгоритмов.

Ключевые слова: манипуляционный робот, планирование пути, алгоритм деления ячеек.

Рабочая среда манипулятора может иметь различную степень определенности. Если в рабочей среде находятся статические препятствия, информация о которых неполна или отсутствует, то в этом случае говорят об управлении манипулятором в неизвестной статической среде. В различных областях применения манипуляционных роботов типичной является задача поиска пути из заданного начального положения в заданное конечное положение робота. Предварительно заданы начальная конфигурация манипулятора $q^0(q_1, q_2, \dots, q_n)$; множество целевых конфигураций Q^T , где $q_i^T \in Q^T$ обеспечивают заданное положение схвата, T – время пути. Найти путь – значит найти такую последовательность конфигураций манипулятора $P = \{t|q(t)\}$, где $t \in [0; T]$ и q – конфигурация манипулятора, чтобы в любое время t ни одна из точек, принадлежавших манипулятору, не налегала на препятствия, а $P(0) = q^0$ и $P(T) = q^T$, т. е. требуется за конечное число шагов передвинуть манипулятор из начальной конфигурации в целевую, обходя препятствия. Целевое состояние может задаваться одной конкретной конфигурацией манипулятора. Предполагается, что имеется априорное знание о том, что она не налегает на препятствия, допустима с точки зрения конструктивных возможностей манипулятора и достижима.

Манипуляционный робот является разновидностью динамических систем. В [2] рассмотрен алгоритм управления динамическими системами, в дальнейшем будем называть его АНС (алгоритм для неизвестной среды). Предполагается, что динамическая система функционирует в n -мерном пространстве состояний, в ее рабочей среде могут присутствовать неизвестные запрещенные статические состояния. Также в статье представлен обзор работ, посвященных алгоритмам управления различными динамическими системами в неизвестной среде, приводятся сравнительные характеристики алгоритмов.

В [12] описан алгоритм создания карты и планирования пути для автономного мобильного робота в неизвестной статической среде. Робот изучает среду с помощью вероятностной интерпретации, основанной на видеосенсорах, видимых углов и границ. Робот перемещается в пространстве, ориентируясь на основе визуального сходства, т. е. ассоциативно. Для построения ассоциативной памяти применяется нейронная сеть. Робот получает от видеокамеры изображение,

которое поступает в виде вектора на входной слой нейронной сети, с помощью которой текущее изображение сравнивается с уже имеющимися изображениями и вставляется в сегмент с наиболее похожими. Таким образом, среда может быть изучена и представлена, как серия визуально различных местоположений. Топология среды строится роботом во время планирования пути путем выявления общих свойств у известных местоположений. Данный алгоритм планирования подходит для двумерного пространства состояний. В [8] представлен алгоритм планирования локального пути для мобильного робота с видеосенсорной системой и метод трансформации двумерных изображений в трехмерную модель. Задача решается в три этапа. На первом этапе детекторы обнаруживают препятствия. На втором этапе двумерное изображение препятствий трансформируется в трехмерную модель, и на третьем – планируется локальный путь при помощи алгоритма фронта распространения волны, избегающий столкновения с этими препятствиями. В [14] представлен алгоритм планирования пути мобильного робота в неизвестной среде. На пространство накладывается двумерная сетка, затем датчики робота определяют, в какой из ячеек сетки находятся препятствия, т. е. строится следующая модель пространства – ячейкам присваиваются различные значения в зависимости от того, есть ли в них препятствия или нет, или через сколько времени препятствие будет в эту ячейку. После чего планирование пути по сетке пространства осуществляется при помощи генетического алгоритма.

Планирование пути манипулятора усложняется многозначностью манипуляторов, в [4] разработан метод, подходящий для планирования пути избыточных и гиперизбыточных манипуляторов: ВВ-метод (ВВ-method). Путь без столкновений с препятствиями находится с помощью постепенных изменений первоначального пути, который обычно является прямой, связывающей начальное и целевое положения. Сложность алгоритма зависит линейно от степеней свободы робота. ВВ-метод не гарантирует достижения цели. В [5] представлен метод планирования в неизвестной среде: utility-guided planning («полезно-управляемое планирование»). Планирование utility-guided – итеративный подход, который начинается «без знания мира» и постепенно строит некоторую

аппроксимацию окружающей среды, модель, удобную для планирования. Представленный подход может использоваться как фрейм для методов планирования, и подходит для динамической среды.

Сформулируем требования к методу управления манипулятором в неизвестной среде: метод должен гарантировать достижение цели, подходить для использования видеосенсорной системы, применяться для n -звенных манипуляторов, иметь возможность встраивания различных алгоритмов планирования пути в областях, где препятствия известны. АНС удовлетворяет данным требованиям [2]. Планирование движения динамической системы в пространстве состояний при неизвестных запрещенных состояниях сводится к решению конечного числа задач планирования в пространстве с известными запрещенными состояниями [1; 11].

В АНС информация о запрещенных состояниях поступает по мере движения робота и в связи с этим образуются зоны свободных и запрещенных точек. Таким образом, возникает потребность планирования пути в среде с известными запрещенными состояниями, используя информацию об этих зонах. Наилучшим образом для этой цели подходят алгоритмы разделения ячеек. Так как с помощью алгоритмов разделения ячеек планирование происходит по свободным областям, получается, что основной единицей вычислений в алгоритме является множество точек, это позволит сократить время планирования пути и при реализации алгоритма сократить объем вычислительных ресурсов.

Различные виды алгоритмов точного разделения ячеек представлены в [10; 13]. *Вертикальное разделение ячеек* или *трапецевидная декомпозиция* для двумерного и трехмерного пространства, описана в [6; 7; 9; 10; 13], в [10] так же указано, что метод может быть усовершенствован для применения в n -мерном пространстве. Однако препятствия должны быть кусочно-линейными, т. е. должны быть представлены как полуалгебраическая модель [10], для которой все примитивы являются линейными. К сожалению, в большинстве задач планирования движения возникают нелинейные алгебраические примитивы из-за нелинейных преобразований, которые появляются в результате вращений форм.

Метод *триангуляции пространства* для двумерного пространства, описанный в [6; 7; 10], заключается в представлении свободного пространства в виде симплицеального комплекса. Алгоритм триангуляции очень сложен и затрачивает много времени на исполнение.

Цилиндрическая декомпозиция [10] похожа на вертикальное разделение ячеек, за исключением того, что границы разделения ячеек не останавливаются на границах препятствий, а расходятся в бесконечность. Получаем точки, которые принадлежат как свободному пространству, так и препятствиям. Для окончательного описания границ ячеек используются логические предикаты. От цилиндрической декомпозиции легко перейти к любой размерности пространства,

топологии пространства конфигураций и полуалгебраическим моделям. Цилиндрическая алгебраическая декомпозиция решает любую задачу планирования движения. Однако определение смежности ячеек для данного метода является трудной задачей. Время работы цилиндрической алгебраической декомпозиции зависит от многих факторов, но основное влияние на длительность оказывает количество полиномов в модели.

Также в [13] представлены *декомпозиция бустрофедон* (*boustrophedon decomposition*), для усовершенствования данного метода – *декомпозиция Морзе* и *декомпозиция, основанная на видимости* (*Visibility-based decomposition*), которые применяются в двумерном пространстве.

Существуют методы *аппроксимирующей декомпозиции* [9]. В них также свободное пространство представлено в виде множества ячеек, но ячейки имеют простую заданную форму, например прямоугольники.

Большинство работ рассматривают алгоритмы разделения ячеек для двумерного пространства [10; 13], такие алгоритмы больше направлены на генерацию пути для мобильных роботов, и не удовлетворяют поставленным задачам в управлении манипуляторами, имеющими более двух степеней подвижности.

Ниже предлагается версия алгоритма разделения ячеек (АРЯ), подходящего для генерации пути движения n -звенного манипулятора, который объединяет разрешенные конфигурации манипулятора в «свободные области», сокращая этим время вычисления пути.

Алгоритм разделения ячеек. Полное описание алгоритма дано в [3]. Планирование в известной среде осуществляется в дискретизированном пространстве обобщенных координат. *Ячейкой* назовем множество точек дискретизированного пространства, находящихся на одной линии дискретизации и имеющих координаты (x_1, x_2, \dots, x_n) , где, например, x_1, x_2, \dots, x_{n-1} имеют одинаковые значения для всех точек одной ячейки, а x_n – различны. Хотя данный способ выделения ячеек не объединяет точки пространства в ячейки во всех измерениях, но даже такая декомпозиция позволит нам оперировать множеством точек одновременно и ускорить работу алгоритма планирования пути. *Свободной областью* называется множество смежных точек ячейки, не содержащих ни одной запрещенной точки.

Алгоритм имеет три этапа и вызывается из АНС каждый раз, когда возникает необходимость в планировании предварительного пути. Исходные данные: $X^0 = q^n$, $X^T = q^T$, Q (множество запрещенных точек, включая конструктивные ограничения), где q^n – текущая конфигурация манипулятора, точка смены пути [2].

Этап 1. Создание таблицы свободных областей *TFR* (*table of free regions*) на основе информации, переданной сенсорной системой при вызове подпрограммы планирования пути (используется множество Q), если она еще не создана. Если таблица TFR уже существует, то в нее вносятся изменения в соответствии с новой информацией, поступившей от сенсоров. В каждой ячейке может быть одна или более свободных

областей, i -ая строка TFR соответствует i -ой ячейке, каждая свободная область этой ячейки характеризуется двумя крайними точками. По определению ячейки, компоненты координат точек, принадлежащих одной ячейке, одинаковы, кроме одной компоненты x_n , значение этой компоненты и будет служить уникальным идентификатором каждой точки в одной ячейке. Так как в свободной области точки являются смежными, то для обозначения границ области достаточно указать идентификаторы двух крайних точек (табл. 1).

Этап 2. Построение графа поиска, который представляет собой дерево соединений свободных областей. Корнем дерева является свободная область, содержащая целевую конфигурацию X^T . Узел дерева имеет как входящие, так и исходящие ветви. Корень дерева – это узел, у которого только исходящие ветви. Лист дерева – это узел, у которого только входящие ветви. Значениями узлов дерева являются границы свободных областей ячеек. Лист присоединяется к узлу, если его свободная область соединима со свободной областью данного узла (определяется по TFR) и если такого узла еще нет в дереве. Построение дерева прекращается, если листом дерева стала область, содержащая начальную конфигурацию X^0 . Для выбора следующего рассматриваемого листа использовался алгоритм поиска в глубину.

Этап 3. Строим путь, используя дерево соединения свободных областей, поднимаясь от последнего присоединенного листа, в котором указана свободная область, содержащая начальную точку X^0 , к корню дерева, в котором содержится область с конечной конфигурацией X^T . Путь строится в 3 этапа:

3.1. Строим последовательность базовых точек пути, выбирая по одной точке из каждой свободной области, через которые будет проходить путь, т. е. области, принадлежащие узлам дерева между последним узлом и корнем.

X^0 – первая точка последовательности базовых точек пути.

При построении базового пути используется уравнение прямой, проходящей через две точки на плоскости:

$$\frac{x - x_1}{x_2 - x_1} = \frac{y - y_1}{y_2 - y_1}. \quad (1)$$

Из формулы (1) получим

$$y = \frac{y_2 - y_1}{x_2 - x_1} \cdot x + \left(y_1 - x_1 \cdot \frac{y_2 - y_1}{x_2 - x_1} \right). \quad (2)$$

Задавая абсциссу x точки, получаем $y(x_1, y_1)$ – координаты предыдущей точки базового пути, (x_2, y_2) – координаты целевой точки. Так как в ячейке все точки имеют одинаковые компоненты координат, кроме компоненты x^n , используем проекции предыдущей точки базового пути и целевой точки на плоскость Ox_1x_n как точки для прямой на плоскости и находим компоненту x^n . Для пути выбираем ближайшую к (x, y) разрешенную точку рассматриваемой свободной области.

3.2. Добавляем точки в последовательность базовых точек пути, если между двумя точками расстояние больше одного шага дискретизации.

Просматриваем последовательность базовых точек и находим пару точек, расстояние между которыми больше одного шага дискретизации. Запишем координаты двух соседних точек последовательности в следующем виде (на примере трехмерного пространства):

$$(x_1^1, x_2^1, x_3^1) \text{ AND } (x_1^2, x_2^2, x_3^2). \quad (3)$$

Если $|x_3^1 - x_3^2| > 1$, то расстояние между точками больше одного шага дискретизации. Добавим столько точек между ними, сколько необходимо. Найдем x_3 первой точки между этими двумя точками:

$$\text{если } x_3^1 < x_3^2, \text{ то } x_3 = x_3^1 + 1;$$

$$\text{если же } x_3^1 > x_3^2, \text{ то } x_3 = x_3^1 - 1.$$

Остальные компоненты координат берем, для начала, как у предыдущей точки. Проверяем, принадлежит ли новая точка той же свободной области, что и предыдущая точка последовательности базовых точек. Если точка не лежит в границах области, тогда компоненту x_3 новой точки оставляем той же, но оставшиеся компоненты координат берем, как у следующей точки, из последовательности базовых точек и т. д.

3.3. Удаляем петли из пути, т. е. отрезки пути, которые выходят из какой-либо точки, а затем в нее же возвращаются или проходят в одном шаге дискретизации. Последовательно перебираем точки пути и для каждой точки ищем среди оставшихся точек последовательности точку, которая равна текущей или лежит в одном шаге дискретизации от текущей. Если такая точка найдена, то все точки от текущей до найденной, включая найденную точку, удаляются из пути.

После того как предварительный путь получен, управление возвращается АНС.

Таблица 1

TFR

№ ячейки	Количество свободных областей	Границы области 1	Границы области 2	...	Границы области M
1	1	$x_n(1) - x_n(k)$...	
2	M	$x_n(1) - x_n(\alpha_1)$	$x_n(\alpha_2) - x_n(\alpha_3)$...	$x_n(\alpha_j) - x_n(k)$

Примечания: k – количество возможных различных значений компоненты x_n ; $M < k$ – количество свободных областей в ячейке; $\alpha_1, \alpha_2, \dots, \alpha_j \in [x_n(1); x_n(k)]$ – какая-то компонента координат x_n .

Программная реализация. На основе алгоритмов создано программное обеспечение управления манипулятором в среде с неизвестными препятствиями.

Задана рабочая сцена для тестирования (рис. 1 и 2).

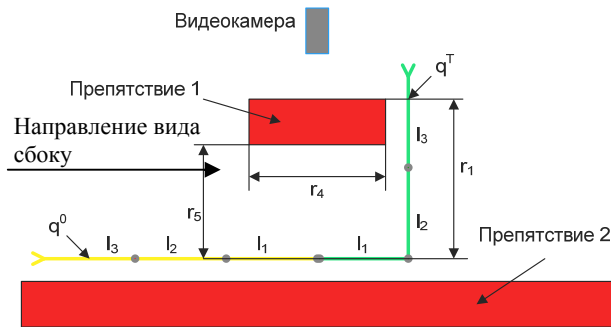


Рис. 1. Рабочая сцена (вид сверху)

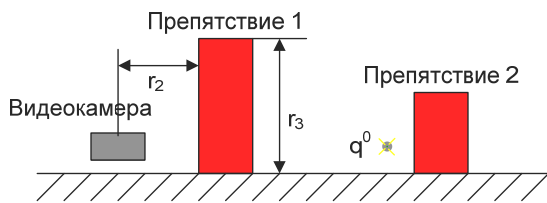


Рис. 2. Рабочая сцена (вид сбоку)

Видеокамера определяет расстояния r_2 , r_3 и r_4 . Тестовая рабочая сцена удовлетворяет следующим соотношениям: $l_1 < r_5$, $l_2 < r_5$, $l_3 < r_5$, $l_2 + l_3 > r_1$, где l_1, l_2, l_3 – длины звеньев манипулятора.

Начальная и целевая конфигурации (рис. 3) заданы оператором в виде векторов обобщенных координат: $q^0 = (3,14; 0; 0)$ радиан, $q^T = (0; 1,57; 0)$ радиан, шаг дискретизации равен 0,314 радиан, количество шагов

дискретизации – 20. Оператором заданы также вектора ограничений, наложенные на звенья, вектор нижних значений $X^L = (0; -3,14; -3,14)$, вектор верхних значений $X^U = (6,28; 3,14; 3,14)$. Препятствия – два параллелепипеда темного цвета, размер и положение которых заданы оператором, но до начала движения подсистеме планирования пути о препятствиях ничего не известно. Роботу поставлена задача переместиться из конфигурации q^0 в конфигурацию q^T , $q^n = q^0$.

Перед началом планирования пути модуль имитации видеосенсорной системы обнаруживает некоторые препятствия в рабочей зоне. Камера (см. рис. 1 и 2), «видит» только ближайшую к камере грань малого препятствия. Модуль рассчитывает запрещенные конфигурации, которые пересекаются с данной гранью препятствия (см. рис. 3), облако точек – это полученные запрещенные конфигурации манипулятора (рис. 4).

После того, как информация, полученная от видеосенсорной системы, представлена в виде запрещенных конфигураций, алгоритм разделения ячеек генерирует предварительный путь, соединяющий X^0 и X^T (где $X^0 = q^n$, $X^T = q^T$), и возвращает управление АНС.

Конфигурации, которые принимал манипулятор во время движения, отражены на рис. 5 (темного цвета). Путь P состоит из 34 конфигураций, не включая начальную и конечную конфигурации, этот же путь представлен в пространстве обобщенных координат (рис. 6, точки темного цвета).

Путь P соединяет q^0 и q^T и является результатом объединения путей P_i , созданных при каждом вызове подпрограммы планирования пути:

$$P = \bigcup_{i=1}^m P_i, \quad (4)$$

где m – количество вызовов алгоритма разделения ячеек.

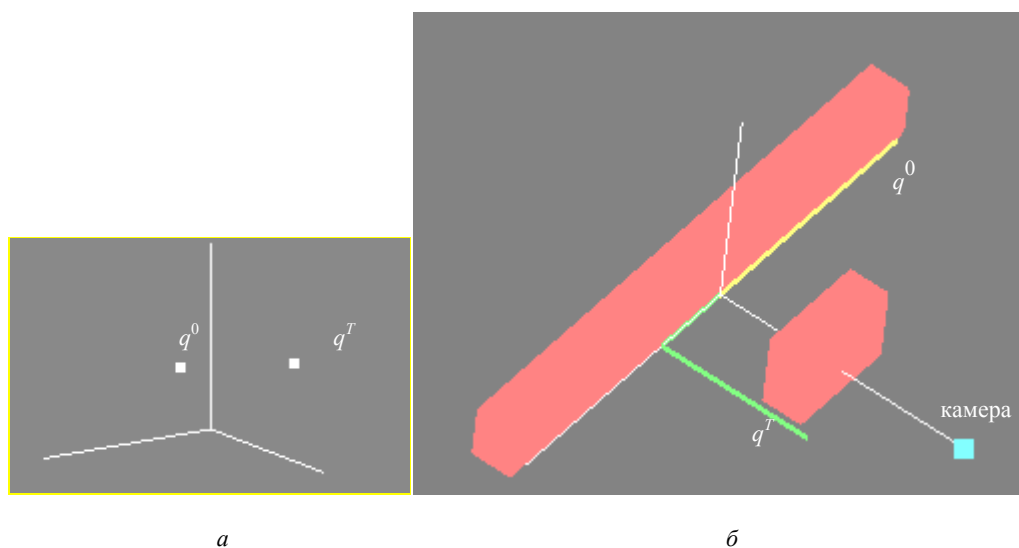


Рис. 3. Стартовая q^0 и целевая q^T конфигурации (а), декартовое пространство (б)

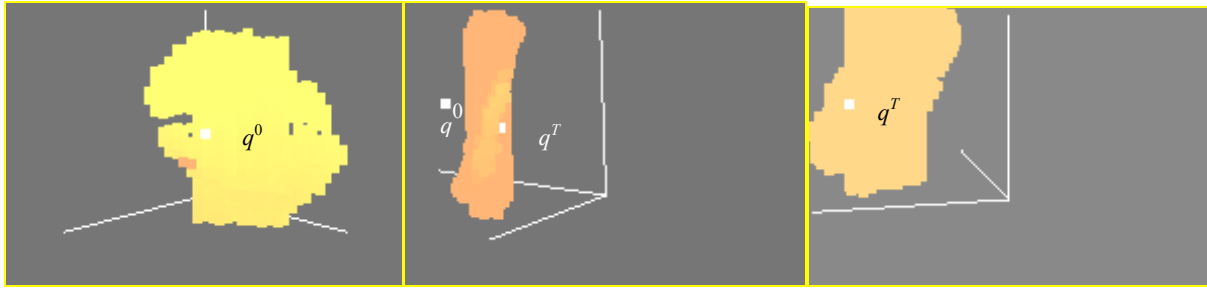


Рис. 4. Расчет запрещенных конфигураций манипулятора

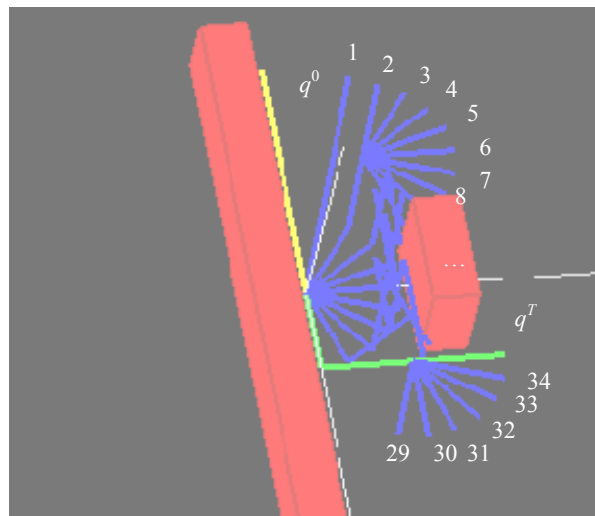


Рис. 5. Путь манипулятора

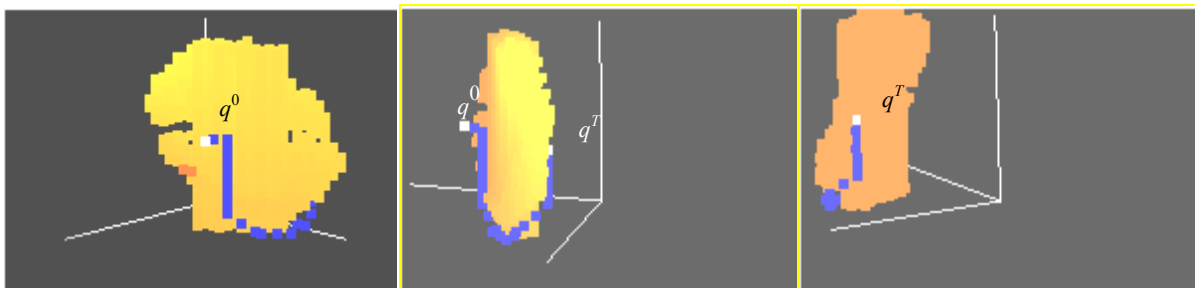


Рис. 6. Конфигурации пути в системе обобщенных координат

Таблица 2

Сравнение времени работы в зависимости от типа сенсорной системы

Количество шагов дискретизации	Не учитывая информацию от видеосенсоров		Учитывая информацию от видеосенсоров	
	Время, с.	Количество вызовов	Время, с.	Количество вызовов
20	228	213	12	10

Данный путь P составлен из путей P_i ($i = 1, 2, \dots, 10$), сгенерированных в результате десяти вызовов подпрограммы планирования пути. Следовательно, в данном примере манипулятор 9 раз столкнулся с ранее неизвестным препятствием, которым являлась

грань препятствия, невидимая для видеокamеры. Это препятствие манипулятор обнаружил имитацией тактильных датчиков.

В случае использования видеосенсорной системы, результат получить значительно быстрее (табл. 2).

Библиографические ссылки

1. Ильин В. А. Интеллектуальные роботы: Теория и алгоритмы / Сиб. аэрокосмич. акад. Красноярск, 1995.
2. Лопатин П. К. Алгоритм управления динамическими системами в неизвестной статической среде // Мехатроника. Автоматизация. Управление. 2007. № 2. С. 9–13.
3. Сафиуллина Н. Ф. Управление манипуляционными роботами в неизвестной среде с использованием разделения областей: выпускная квалификационная работа на соискание степени магистра техники и технологий / Сиб. гос. аэрокосмич. ун-т. Красноярск, 2008.
4. Baginski B. Motion Planning for Manipulators with Many Degrees of Freedom – The BB-Method [Electronic resource] : phys. dr dissertation / Munich Technical University, 1998. URL: books.google.ru/books.
5. Burns, B. Exploiting Structure: Guided Approach to Sampling-Based Robot Motion Planning [Electronic resource] : phys. dr dissertation / Graduate School of the University of Massachusetts. URL: <http://ieeexplore.ieee.org/iel5/4209048/4209049/04209602.pdf>, 2007.
6. Chazelle B. Approximation and decomposition of shapes // Algorithmic and Geometric Aspects of Robotics / ed. J. T. Schwartz, C. K. Yap ; Lawrence Erlbaum Associates. Hillsdale, NJ, 1987. P. 145–185.
7. Chazelle B. Triangulating a simple polygon in linear time // Discrete Comput. Geom. 1991. № 6(5). P. 485–524.
8. Wavefront Method-Based Local-Path Planning for a Mobile Robot with a Vision System / J.-W. Kwon, D.-H. Yang, D. Chwa, S-K. Hong // SICE-ICASE Intern. Joint Conf. (18–21 Oct., 2006, Busan). Busan, 2006. P. 1250–1254.
9. Latombe J. C. Robot motion planning. Boston : Kluwer Academic Publishers, 1990.
10. LaValle S. M. Planning Algorithms [Electronic resource]. Cop. 1999–2006. URL: <http://msl.cs.uiuc.edu/planning>.
11. Lopatin P. K. Algorithm of a manipulator movement amidst unknown obstacles // Proc. of the 10th Intern. Conf. on Advanced Robotics (ICAR2001) (22–25th of August, 2001, Hungary). Hungary, 2001. P. 327–331.
12. Meikle S., Yates R., Harris A. Computer Vision Algorithms for Autonomous Mobile Robot Map Building and Path Planning // ITSC'97 : IEEE Conf. on Intelligent Transport. System (9–12 Nov. 1997, Boston). Boston, 1997. P. 99–109.
13. Principles of robot motion: theory, algorithms and implementation / H. Choset [et al.]. Cambr. dge : A Bradford book, 2005.
14. Zhang Y., Zhang L., Zhang X. Mobile Robot Path Planning base on the Hybrid Genetic Algorithm in Unknown Environment // ISDA'08 : VIII Intern. Conf. on Intelligent Systems Design and Applic. (26–28 Nov. 2008, Kaohsiung). Kaohsiung, 2008. P. 661–671.

N. F. Safiullina, P. K. Lopatin

CELL DECOMPOSITION ALGORITHM FOR MANIPULATORS PATH PLANNING

The authors use the well known approach to manipulators path planning in unknown environment, which is reduced to solution of final number path planning tasks in known environment. The new version of cell decomposition algorithm is applied to path planning in known environment in n-dimensional configuration space of generic coordinates. The results of the algorithms implementation are presented.

Keywords: manipulator, cell decomposition algorithm, path planning.

© Сафиуллина Н. Ф., Лопатин П. К., 2012

УДК 621.89.-229.3

Л. В. Строк, В. С. Секацкий, Я. Ю. Пикалов, М. В. Брунгардт

РАСЧЕТ ХАРАКТЕРИСТИК НЕЗАМКНУТОГО ОСЕВОГО ГИДРОСТАТИЧЕСКОГО ПОДШИПНИКА С УЧЕТОМ ШЕРОХОВАТОСТИ ОПОРНЫХ ПОВЕРХНОСТЕЙ

Разработана математическая модель и проведены исследования влияния шероховатости опорных поверхностей на характеристики незамкнутого осевого гидростатического подшипника с дроссельной компенсацией расхода смазки. Показано влияние высоты и шага шероховатости на расходные и нагрузочные характеристики.

Ключевые слова: гидростатические подшипники, математическая модель, шероховатость опорных поверхностей.

Качество конечного изделия зависит от качества проектирования, в частности, от точности и достоверности расчетов основных параметров изделия. Особенно это актуально при проектировании гидростати-

ческих опор (подшипников и направляющих). Традиционные методики расчета гидростатических опор основаны на использовании в качестве исходных параметров их номинальных значений. Однако откло-