

УДК 681.5.015

**АЛГОРИТМ УПРАВЛЕНИЯ ДВИЖЕНИЕМ РОБОТА  
В УСЛОВИЯХ НЕДЕТЕРМИНИРОВАННОЙ ИНФОРМАЦИИ О ВНЕШНЕЙ СРЕДЕ****А.В. ГОСПОД***(Могилевский государственный университет продовольствия)*

*Планирование движения робота в условиях недетерминированной информации о внешней среде часто принимает допущения об известности геометрии препятствия и возможности предсказать траекторию движения экстраполяцией, что в реальности невозможно. Рассматривается алгоритм collision-free perceiver (CFP – бесконтактная воспринимающая), который может обнаружить траектории, свободные от столкновений при известных геометрии или движении. Анализируется, как использовать CFP в режиме реального времени на роботе с  $n$  степенями свободы при неизвестных траектории и размерах препятствий, минимизируется количество опасных остановок, когда робот мог бы столкнуться с объектом.*

**Введение.** Планирование движения роботов в неопределенной динамичной среде привлекает все больше внимания в исследовательской робототехнике. В [1] был введен алгоритм collision-free perceiver (CFP – бесконтактная воспринимающая), который может обнаружить траектории, свободные от столкновений при известных геометрии или движении. Одно из известных предположений, ничего не имеющее общего с реальностью, о том, что движение препятствия, известно [2; 3]. Второе предположение – определена геометрия препятствия. Если движение неизвестно, то используется общий подход для прогнозирования будущего движения, отслеживая прошлое [4–9]. Таким образом, на столкновение робота с конфигурацией  $q$  в будущем времени  $t$  указывает проверка пересечения конфигурации робота  $(q, t)$  с каждой частью препятствия в момент времени  $t$ .

Существует много быстрых алгоритмов проверки столкновений [10–12], которые могут быть эффективно использованы для решения вопроса ограничения числа препятствий. Известны алгоритмы для планирования движения мобильного робота [13–15] и для мобильного движения манипулятора [16]. Тем не менее они требуют либо известную геометрию, либо быструю и точную сегментацию объектов, что очень сложно и практически невозможно в динамических средах. Кроме того, предсказание может быть достаточно точным только в течение короткого периода времени, т.е. только сразу после того, как оно сделано. Для компенсации этого требуется частое повторное прогнозирование и расчет проверки столкновения. Предполагая известность геометрии препятствий, алгоритм на самом деле предполагает очень быстрое и точное распознавание объекта.

Есть прогресс в обнаружении препятствий для мобильных роботов, таких как бездорожье и постройки [17; 18], растительность [19], люди [20]. Тем не менее в многолюдных помещениях с множеством неизвестных параметров обнаружение всех объектов слишком медленное и неточное, а также без надобности. Например, представьте банкет с большим количеством людей в небольшом помещении, где робот, подавая напитки и угощения, маневрирует с тарелками между присутствующими. Невозможно просчитать все. С другой стороны, все и не надо, достаточно доставить необходимое, избегая столкновений. Таким образом, необходимо изучить, как планировать движение робота без необходимости просчета всех препятствий, которые могут двигаться и в неизвестных направлениях.

В [1] были введены две новые концепции, которым не нужны ни геометрия, ни траектория препятствий: а) точечные препятствия (ТП), представляют недетерминированную среду непосредственно из данных датчиков низкого уровня в каждый момент времени; б) динамический пакет (ДП), обнаруживающий конфигурацию робота в момент времени  $(q, t)$ , которая гарантированно будет свободна от столкновений в промежутке времени  $[t, t)$ . Также предположено, что препятствие движется с любой скоростью в интервале  $[0, v_{\max}]$ . Затем ввели алгоритм [21] для проверки пересечения динамического пакета и точечного препятствия, а также методику [22], используемую для анализа непрерывности свободной траектории, проверяющую, гарантируется ли специальный набор дискретных точек без столкновения.

Изложенный выше подход, который называется «бесконтактная воспринимающая (CFP)», позволяет планировать движение робота в условиях недетерминированной информации о внешней среде в реальном времени. Тем не менее обнаружение траектории, свободной от столкновений, не может быть мгновенным. И в момент вынужденной остановки из-за препятствия робот может быть сбит другим объектом. Поэтому важным практическим вопросом является минимизация небезопасных остановок, принимая во внимание ограниченное время для обнаружения столкновения. Для решения этого вопроса в CFP применен адаптивный алгоритм реального времени RAMP [16] (расширение RAMP) включающий временные ограничения T-RAMP.

### 1. Обзор CFP и RAMP

Сначала рассмотрим CFP на основе понятий точечных препятствий и динамического пакета, а затем опишем RAMP и как RAMP можно включить в CFP.

#### 1.1. Обозначения

Следующие обозначения описывают модель робота в декартовых координатах:  $R(q)$  – область, занятая роботом  $R$  в  $R^3$  конфигурации  $q$ .

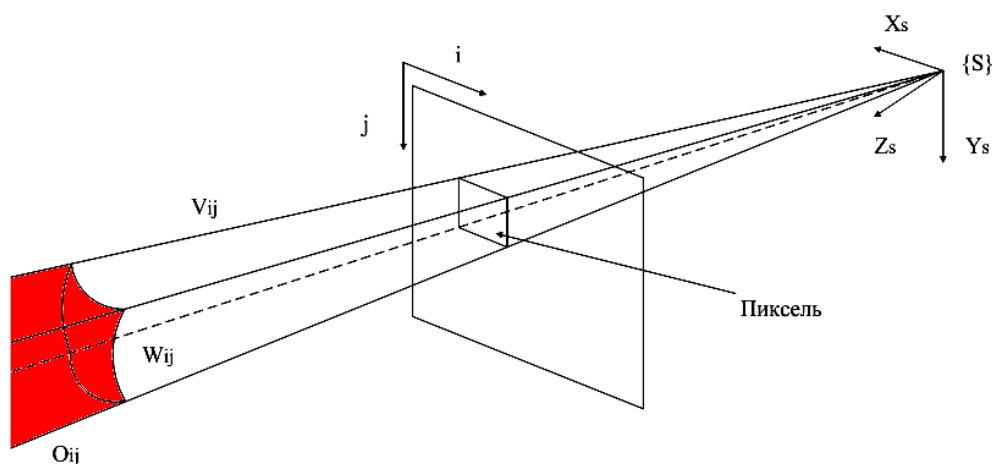
Также используем различные временные обозначения:  $\tau$  – время зондирования;  $t$  – время проведения операции.

Обозначим верхнюю границу скорости для всех объектов  $v_{\max}$ .

#### 1.2. Точечные препятствия

Точечные препятствия обнаруживаются напрямую при мгновенном зондировании датчиком низкого уровня, без обработки данных. Образ  $I(\tau)$  от линии прямой видимости датчика, например, лазерного дальномера, гидролокатора, стереовидения и т.д., каждый пиксель  $(i, j)$  сопоставляется с точкой  $(x, y, z)$  на физическом объекте и  $d_{ij}$  расстояние от  $(x, y, z)$  до начала координат  $\{S\}$  кадра датчика. Пусть  $W_{ij}$  пересечение объема наблюдаемого квадратного пикселя  $(i, j)$  (пиксель определяется четырьмя углами с лучами из  $\{S\}$ ) и сферой с центром в  $\{S\}$  с радиусом  $d_{ij}$ . Точечное препятствие  $O_{ij}$  формируется  $W_{ij}$  (рисунок).

Общность ТП в каждый конкретный момент  $\tau$  может быть рассмотрена как сформированное пространство препятствий  $O_s(\tau)$ . Точечные препятствия одного момента не связаны с ТП другого момента времени.



Геометрия точечного препятствия  $O_{ij}$  от линии прямой видимости датчика

#### 1.3. Динамический пакет

В некоторый момент времени  $\tau_0$  мы стремимся обнаружить будущую точку  $\chi = (q, t)$  при  $t > \tau_0$ , свободную от столкновений. Для этого необходимо наблюдать за изменениями окружающей среды во временном интервале  $[\tau_0, t]$ , как перемещаются препятствия относительно  $R(q)$ , чтобы обнаружить точку  $\chi$ , свободную от столкновений за время  $\tau < t$ , или точку  $\chi$  столкновения за время  $\tau = t$ .

Динамический пакет  $E(\chi, \tau)$  как функция, зависящая от времени  $\tau \leq t$ , представляет собой замкнутую поверхность, вмещающую области  $R(q)$  в трехмерном пространстве  $R^3$  (или  $R^2$  для плоскости), такие, что минимальное расстояние от любой точки на  $E(\chi, \tau)$  и  $R(q)$ :

$$d(t, \tau) = v_{\max}(t - \tau). \tag{1}$$

Таким образом,  $E(\chi, \tau) = R(q) \oplus B(t, \tau)$ , где  $B(t, \tau)$  представляет собой шар с центром в начале координат и диаметром  $d(t, \tau)$ .

Динамический пакет обладает следующими свойствами:

- 1) монотонно уменьшается с течением времени со скоростью  $v_{\max}$ , т.е.  $E(\chi, \tau_{i+1}) \subset E(\chi, \tau_i)$ , где  $i > 0$ ,  $\tau_i < \tau_{i+1} \leq t$ .  $E(\chi, \tau)$  сжимается до  $R(q)$  при  $t$ ;
- 2) текущее препятствие, которое находится на поверхности или внутри  $E(\chi, \tau_i)$ , никогда не будет находиться снаружи или внутри  $E(\chi, \tau_{i+1})$ ;
- 3) текущее препятствие, находящееся на поверхности или внутри  $E(\chi, \tau_i)$ , может быть вне  $E(\chi, \tau_i)$ , для некоторых  $\tau_j \in (\tau_i, t]$ , если нет движения в направлении  $R(q)$  с максимальной скоростью  $v_{\max}$ .

#### 1.4. Алгоритм CFP

Этот алгоритм находит точку  $\chi = (q, t)$ , свободную от столкновений, проверяя пересечения между динамическим пакетом  $E(\chi, \tau)$  и точечным препятствием в каждый момент времени, начиная с  $\tau_0$  и до

любого столкновения, или за максимальное время наблюдений  $t - \tau_0$ , который контролируется системной переменной тактового сигнала  $t_{\text{clock}}$  (эта переменная изменяется независимо от алгоритма CFP), как показано в алгоритме 1. Интервал между двумя соседними моментами зондирования  $\sigma\tau$ , т.е. частота зондирования  $1/\sigma\tau$ . Каждая итерация цикла, как правило, занимает больше времени, чем  $\sigma\tau$ , следовательно, после каждой итерации есть обновленные данные от датчика для следующей итерации.

Алгоритм 1. Восприятие без столкновений CFP

```

1: ввод  $\chi = (q, t)$ ,  $\tau = \tau_0$ ,  $\sigma\tau$ ,  $t_{\text{clock}} = 0$ 
2: получить динамический пакет  $E(\chi, \tau)$ 
3: while  $((\tau < t)$  и  $(t_{\text{clock}} < t - \tau_0))$  do
4:     if  $E(\chi, \tau)$  не пересекает точечные препятствия в момент  $\tau$  then
5:         изменить  $E(\chi, \tau)$ 
6:         получение  $\chi$ , свободной от столкновений
7:     else
8:          $\tau = \tau + \sigma\tau$  (для следующих данных от датчика)
9:     end if
10: end while
11: получение  $\chi$ , не свободной от столкновений

```

Как правило, для робота, состоящего из нескольких звеньев, каждое звено можно аппроксимировать набором геометрических фигур, таких как прямоугольник, квадрат, сфера, параллелограмм. Теперь ДП может быть создан для каждого звена. Так как ДП для всего робота – сумма ДП звеньев, необходимо сосредоточиться на том, как проверить пересечение между ДП звеньев и ТП. Используем экстракцию, чтобы определить ТП. В [21] описана проверка пересечения ДП и ТП (IDEAOS), наиболее эффективный алгоритм, заявленный в CFP.

### 1.5. Адаптивный алгоритм реального времени RAMP

Парадигма RAMP [16] мотивирует планировать движение в режиме реального времени для роботов с большим количеством степеней свободы, таких как мобильные манипуляторы, в динамических средах с неизвестной траекторией движения препятствий. Известно, что никакой алгоритм, планирующий движение для роботов с большим количеством степеней свободы, не выполним даже для определенной и статической окружающей среды из-за огромной проблемы построения многомерных  $S$ -препятствий. Таким образом, широко используются планировщики, основанные на дискретизации, в частности PRM [23] и RRT [24].

Алгоритм RAMP также основан на дискретизации, это особенно эффективно при планировании движения робота с большим количеством степеней свободы в условиях недетерминированной информации о внешней среде из-за следующих особенностей:

- в реальном времени одновременное движение робота и планирование его траектории;
- параллельное планирование с оптимизацией, как в эволюционном алгоритме [25], и постоянное обновление возможных траекторий от его текущей конфигурации к целевой;
- структурная гибкость, чтобы позволить онлайн адаптацию к различным внешним условиям и офлайн дополнения для различных конфигураций роботов.

Все основные компоненты RAMP настраиваемы. Сила RAMP в его гибкости для адаптаций и расширений; RAMP всегда имеет набор различных траекторий в пространстве – пакет траекторий. Начальный пакет траекторий может быть сформирован случайно. Каждая траектория начинается с текущей конфигурации робота, заканчивается в целевой и может быть лишь частично выполнимой – имеет неосуществимый сегмент. Качество траектории с точки зрения выполнимости и оптимальности оценено через функцию оценки, объединяющую критерии оптимизации, такие как кратчайшее время движения, максимальное время выполнимого сегмента и т.д.

Как только имеется выполнимый сегмент траектории, RAMP позволяет роботу движение по нему, при этом вычисляя последующие возможные сегменты, таким образом робот может переключиться на более лучший вариант. Три цикла запускаются одновременно:

- сенсорные данные обновляются при каждом зондировании;
- изменение траектории и оценка (переоценка) на основе данных датчика в каждом цикле расчета траектории;
- робот переключается с текущей на лучшую траекторию в цикле контроля или адаптации.

RAMP определяется эффективным онлайн обнаружением выполнимых траекторий. Первоначальный RAMP предполагает, что геометрия препятствий известна, и проводит проверку на основе предсказания движения. Это реально только при моделировании.

### 2. Алгоритм T-RAMP

Используя CFP в RAMP для проверки сегмента траектории (посредством обнаружения свободных от столкновений), можно устранить нереалистичное предположение об известности геометрии препятствия и недостаток в прогнозировании траектории движения препятствия. Сегмент траектории, свободный

от столкновений, найденный CFP, по которому движется робот, обозначается  $\Gamma_1$ , а в это время RAMP ищет следующую возможную траекторию. После того как робот прошел сегмент  $\Gamma_1$ , если найден следующий безопасный сегмент траектории  $\Gamma_2$ , он продолжает плавное движение по нему.

И все же CFP необходимо конечное время, чтобы обнаружить свободные от столкновений точки, по которым пройдет траектория движения робота. Время обнаружения свободных точек  $\chi = (q, t)$  зависит от двух факторов:

- 1) размера динамического пакета  $E(\chi, \tau)$ , который зависит от  $v_{\max}(t - \tau)$  и уменьшается со временем  $\tau$ ;
- 2) вычислительных мощностей компьютера и датчиков.

Первый фактор, как правило, доминирует. Пусть  $\tau_0$  – время начала наблюдений и проверки, свободна ли точка  $(q, t)$  от столкновений. Если  $E(\chi, \tau)$  свободен от точечных препятствий при  $\tau_1 < t$ , то CFP необходимо  $(\tau_1 - \tau_0)$  времени, чтобы найти  $\chi$ , свободную от столкновений. Поэтому следующий выполнимый сегмент траектории не может быть найден при завершении роботом текущего сегмента, приводящего к остановке. Во время такой принудительной остановки робот может быть сбит препятствием.

Поэтому важное расширение RAMP состоит в минимизации остановок. Параллелизм и гибкость RAMP добивается следующим:

- добавлением к функции оценки в качестве дополнительного критерия оптимизации, увеличение безопасного времени  $\delta t_{\text{safe}}$ , которое он проводит в сегменте траектории без столкновений. RAMP выберет допустимый сегмент траектории (среди всех найденных возможных), который максимизирует общее время  $\Delta t_{\text{safe}} = \Delta t_{\text{move}} + \delta t_{\text{safe}}$ , где  $\Delta t_{\text{move}}$  – время движения в сегменте. Примечательно, что  $\delta t_{\text{safe}}$  в точке  $\chi_e = (q_e, t_e)$  не может быть известно точно, прежде чем робот достигнет  $\chi_e$ , но его можно оценить с помощью  $d_{\min}(q_e, \tau)/v_{\max} - (t_e - \tau)$  для  $\tau < t_e$ , где  $d_{\min}(q_e, \tau)$  – минимальное расстояние между роботом и ТП. Поскольку  $\delta t_{\text{safe}}$  рассчитывается для точки  $\chi_e$  без столкновений при  $\tau > t_e$ . Разработан метод расчета  $d_{\min}(q_e, \tau)$  между точечными препятствиями вблизи робота в точке  $\chi_e$ ;

- отделением проверки столкновений (CFP) от оценки пригодности траектории. Проверка столкновений постоянно происходит в фоновом режиме и дает информацию о свободной траектории, в то время как блок оценки пригодности просто использует эту информацию для вычисления значения функции пригодности, что гораздо быстрее, чем ранее. Назовем расширенный алгоритм T-RAMP «алгоритм 2».

#### Алгоритм 2. T-RAMP

инициализация исходных данных

$m$  – количество итераций зондирования в цикле планирования

$n$  – количество итераций планирования в цикле адаптации

$\Delta t_{\min}$  – шаг времени

$q_e$  – начальная конфигурация робота

$t_e$  – текущее время  $\tau$  ( $\tau$  – системное автоматически обновляемое время)

инициализация набора траекторий  $S$ , соединяющих начальную конфигурацию робота и конечную

$\Delta t_{\text{move}} = 0$

$\delta t_{\text{safe}} = 0$

while <робот достиг цели> do

одновременно зондирование, проверка столкновений, планирование и адаптация, движение:

зондирование: повторить снятие данных с датчика

планирование на каждой  $m$ -й итерации зондирования или при остановке робота

if  $t_e < \tau$  then

$\Delta t = \max(\Delta t_{\text{move}} + \delta t_{\text{safe}}, \Delta t_{\min})$

$t_e = \tau + \min(mn\delta t, \Delta t)$

установить время начала всех траекторий  $S$  как  $t_e$

end if

изменить  $S$

адаптация при  $\tau = t_e$  или на каждой  $m$ -й итерации зондирования

оценка траектории  $S$

$\Gamma_{\text{best}}$  – лучшая траектория

$q_e$  – последняя конфигурация на сегменте первой траектории без столкновений

$\Delta t_{\text{move}}$  – время, необходимое роботу, для перехода к конфигурации  $q_e$

$t_e - \tau + \Delta t_{\text{move}} + \delta t_{\text{safe}}$

if  $\Delta t_{\text{move}} + \delta t_{\text{safe}} = 0$  then

небезопасный шаг, робот может столкнуться с препятствием

end if

проверка на столкновение: вызов алгоритма 3

движение: перемещение робота по траектории  $\Gamma_{\text{best}}$  за время  $\Delta t_{\text{move}}$

end while

## Алгоритм 3. Проверка на столкновение

1: входные  $S$  траектории  $N$  сегмента, где каждый сегмент траектории  $i$ ,  $0 < i \leq N$ , представляет собой последовательность точек  $\chi_1^i, \chi_2^i, \dots$  продолжительностью  $m \cdot n \cdot \delta t$

2:  $C$  – последовательность точек  $\chi_1^1, \chi_1^2, \dots, \chi_1^N, \chi_2^1, \chi_2^2, \dots, \chi_2^N, \dots$

3: запуск CFP (алгоритм 1) для каждой точки,  $C$  и  $S$  не обновляются

4: результат – точки свободные, от столкновений, найдены для каждой из траекторий; вычислить и передать  $\delta t_{\text{safe}}$

В алгоритме 2 есть четыре одновременных шага: проверка на столкновение, планирование, адаптация и движение. Каждый цикл адаптации состоит из многократных циклов планирования. Каждая итерация цикла зондирования длится  $\delta t$ , которая определяется проектировщиком. Таким образом, новая информация, полученная при каждом зондировании, используется в CFP (алгоритм 1), который в свою очередь вызывается в алгоритме 3 для проверки столкновения.

В каждом цикле адаптации робот одновременно перемещается вдоль допустимой траектории  $\Gamma_{\text{best}}$  и планирует свой следующий возможный сегмент, который начинается с конечной точки  $\chi_e = (q_e, t_e)$  из предыдущего допустимого сегмента  $\Gamma_{\text{best}}$ . Если робот достиг  $\chi_e$ , а следующий сегмент не найден, робот остановится, продолжая поиск возможной траектории. Если время остановки робота меньше  $\delta t_{\text{safe}}$ , значит остановка благополучная, иначе существует опасность столкновения остановившегося робота с другими объектами.

Константы  $m$  и  $n$  зависят от  $v_{\text{max}}$  препятствий и размера окружающей среды.

Следующие функции используются в алгоритме 2, в дополнение к алгоритму 3, для проверки на столкновение:

- инициализация пакета траекторий происходит случайным образом, как в [16], создаются промежуточные конфигурации узла, между начальной и целевой конфигурацией;
- функция оценки значений для траектории, которая максимизирует время движения в сегменте траектории (чтобы обойтись без остановки) и минимизирует общее время движения;
- изменение изначально случайно выбранной траектории  $S$  с помощью добавления, удаления или изменения координат конфигураций или узловых точек, оценка новой траектории и замена на лучшую.

**Заключение.** В результате объединения и улучшения алгоритмов CFP и RAMP получили новый алгоритм T-RAMP, который позволяет планировать движение робота в режиме реального времени и уменьшает количество небезопасных остановок. Полученная методика дает возможность управлять роботом в условиях недетерминированной информации о внешней среде (неизвестные траектория и геометрия препятствий).

## ЛИТЕРАТУРА

1. Vatcha, R. Perceived CT-space for motion planning in unknown and unpredictable environments / R. Vatcha, Xiao, J. // Intl. Workshop on the Algorithmic Foundations of Robotics, WAFR (December 2008).
2. Fiorini, P. Motion planning in dynamic environments using velocity obstacles / P. Fiorini, Z. Shiller // Intl. J. of Robotics Research, 760–772 (1998).
3. Using non-linear velocity obstacles to plan motions in a dynamic environment / F. Large [et al.] // IEEE Intl. Conf. on Control, Automation, Robotics and Vision (ICARCV), pp. 734–739 (2002).
4. Elnagar, A. Motion prediction of moving objects based on autoregressive model / A. Elnagar, K. Gupta // IEEE Trans. on Systems, Man, and Cybernetics (Systems and Humans) 28(6), 803–810 (1998).
5. Chang, C.C. Environment prediction for a mobile robot in a dynamic environment / C.C. Chang, K.-T. Song // IEEE Trans. on Robotics and Automation 13(6), 862–872 (1997).
6. Gatmo: a generalized approach to tracking movable objects / G. Gallagher [et al.] // IEEE Intl. Conf. on Robotics and Automation, pp. 2043–2048 (May 2009).
7. Moving obstacle detection in highly dynamic scenes / A. Ess // IEEE Intl. Conf. on Robotics and Automation, pp. 56–63 (May 2009).
8. Elnagar, A. An adaptive motion prediction model for trajectory planner systems / A. Elnagar, A. Hussein: // Intl. Conf. on Robotics and Automation, pp. 2442–2447 (September 2003).
9. Moving obstacles' motion prediction for autonomous navigation / V. Govea [et al.] // Int. Conf. on Control, Automation, Robotics and Vision (December 2004).
10. I-collide: An interactive and exact collision detection system for large-scale environments / Cohen, J.D. [et al.] // Proc. of ACM Interactive 3D Graphics Conf., pp. 189–196 (1995).
11. Jimenez, P. 3D collision detection: A survey / P. Jimenez, F. Thomas, C. Torras // Computers and Graphics 25, 269–285 (2000).

12. Lin, M.C. Collision detection between geometric models: A survey / M.C. Lin, S. Gottschalk // Proc. of IMA Conf. on Mathematics of Surfaces, pp. 37–56 (1998).
13. Kushleyev, A. Time-bounded lattice for efficient planning in dynamic environments / A. Kushleyev, M. Likhachev // IEEE Intl. Conf. on Robotics and Automation, pp. 1662–1668 (May 2009).
14. High-speed autonomous navigation with motion prediction for unknown moving obstacles / V. Govea [et al.] // IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, pp. 82–87 (October 2004).
15. Van den Berg, J. Anytime path planning and replanning in dynamic environments / J. van den Berg, D. Ferguson, J. Kuffner // IEEE Intl. Conf. on Robotics and Automation, pp. 2366–2371 (May 2006).
16. Vannoy, J. Real-time Adaptive Motion Planning (RAMP) of mobile manipulators in dynamic environments with unforeseen changes / J. Vannoy, J. Xiao // IEEE Trans. on Robotics 24(5), 1199–1212 (2008).
17. Murarka, A. Detecting obstacles and drop-offs using stereo and motion cues for safe local motion / A. Murarka, M. Sridharan, B. Kuipers // IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, pp. 702–708 (2008).
18. Caraffi, C. Off-road path and obstacle detection using decision networks and stereo vision / C. Caraffi, S. Cattani, P. Grisleri // IEEE Trans. on Intelligent Transportation Systems 8(4), 607–618 (2007).
19. Bradley, D. Vegetation detection for driving in complex environments / D. Bradley, R. Unnikrishnan, J.A. Bagnell // IEEE Intl. Conf. on Robotics and Automation (April 2007).
20. Bellotto, N. Multisensor-based human detection and tracking for mobile service robots / N. Bellotto, H. Hu // IEEE Trans. on Systems, Man, and Cybernetics – Part B 39(1), 167–181 (2009).
21. Vatcha, R. An efficient algorithm for on-line determination of collision-free configuration-time points directly from sensor data / R. Vatcha, J. Xiao // IEEE Intl. Conf. on Robotics and Automation (May 2010).
22. Vatcha, R. Perceiving guaranteed continuously collision-free robot trajectories in an unknown and unpredictable environment / R. Vatcha, J. Xiao // IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (October 2009).
23. Probabilistic roadmaps for path planning in high-dimensional configuration spaces / L. Kavraki [et al.] // IEEE Trans. on Robotics and Automation, pp. 566–580 (1996).
24. La Valle, S.M. Planning Algorithms / S.M. La Valle // Cambridge University Press (May 2006).
25. Evolutionary algorithms + domain knowledge = real-world evolutionary computation / P.P. Bonissone [et al.] // IEEE Trans. Evolutionary Computation 10(3), 256–280 (2006).

Поступила 19.09.2014

## ALGORITHM OF THE ROBOT MOTION CONTROL IN THE CONDITIONS OF NON-DETERMINISTIC INFORMATION ABOUT THE EXTERNAL ENVIRONMENT

### A. HOSPAD

*Planning for the robot in a non-deterministic information about the external environment often makes assumptions about the famous geometry constraints and opportunities to predict the trajectory of the extrapolation that in reality impossible. Was introduced algorithm collision-free perceiver (CFP – contactless receptive), which can detect the path free of collision with known geometry and motion. In this paper we consider how to use the CFP in real time, the robot with  $n$  degrees of freedom, with unknown trajectory sizes and obstacles minimizes the amount of hazardous stops when the robot would collide with the object.*