УДК 629.787 DOI: 10.17587/mau.19.734-743

Ван Гуоянь, аспирант, guoyan@yandex.ru,
А. В. Фомичев, канд. техн. наук, доц., a.v.fomichev@bmstu.ru
Московский государственный технический университет им. Н. Э. Баумана, г. Москва

Алгоритм планирования безопасного маршрута движения марсохода с учетом рельефа местности

Для успешного осуществления запланированной миссии полета на Марс необходимо решение проблемы планирования пути движения марсохода по поверхности планеты с учетом особенностей рельефа местности.

Традиционные алгоритмы поиска пути, такие как A, а также улучшенный алгоритм A* — D* и Field D*, которые успешно применялись на планетоходах в состоявшихся экспедициях на Луну и Марс, имеют проблему поиска кратчайшего оптимального пути.*

В то же время такие улучшенные алгоритмы A, как Basic Theta*, Lazy Theta* позволяют планировать движение в любом направлении и искать более короткие маршруты движения.*

В данной статье на основе проведенного сравнительного анализа существующих алгоритмов A, Basic Theta*, Lazy Theta* предложена новая модификация алгоритма планирования движения в любом направлении Lazy AT, позволяющая сократить время вычисления и получить более короткий путь от исходной до конечной точки маршрута.*

На основе анализа поверхностных особенностей рельефа планеты можно судить о показателе опасности ее рельефа. Основной опасностью при движении по поверхности Марса для колесного марсохода являются наклоны и препятствия. Исходя из требований устойчивости положения марсохода при его движении и способности преодоления препятствий были выбраны следующие топографические коэффициенты для описания показателя опасности: угол наклона местности, шероховатость поверхности, размах рельефа.

В новой модификации алгоритма Lazy AT для получения безопасной траектории движения марсохода предлагается использовать индикатор опасности местности (индикатор риска), который учитывается в модификации алгоритма Risk Lazy AT.

Сравнительный анализ результатов моделирования алгоритмов Risk Lazy AT и Lazy Theta показал, что предложенный новый алгоритм Risk Lazy AT может обеспечить безопасное движение подвижного объекта по поверхности планеты.*

Основываясь на реальных данных о рельефе поверхности Марса, была разработана цифровая карта поверхности планеты и проведено моделирование пространственного маршрута движения марсохода с помощью нового алгоритма Risk Lazy AT.

Ключевые слова: *алгоритм A*, алгоритм планирования движения в любом направлении, алгоритм Basic Theta*, алгоритм Lazy Theta*, алгоритм Lazy AT, алгоритм Risk Lazy AT, индикатор опасности местности*

Введение

Одна из важнейших задач, требующих решения при планировании маршрута марсохода, заключается в том, чтобы его траектория движения от начальной до конечной (целевой) точки маршрута, в общем случае проходящая в малоизученной обстановке на поверхности планеты, обеспечивала его гарантированную безопасность. Поэтому разработка алгоритма планирования маршрута марсохода, который в автономном режиме обеспечивал бы поиск оптимальной и безопасной траектории, является в настоящее время весьма актуальной задачей.

Известно, что в уже состоявшихся экспедициях на Луну и Марс на планетоходах успешно применялись следующие подобные алгоритмы: алгоритмы A* и улучшенный алгоритм A*, например D*, а также Field D*, разработанный на базе алгоритма D* [1—5].

Алгоритмы A* и D* являются эвристическими алгоритмами. Как указано в работе [6], эвристический алгоритм (эвристика) — это алгоритм решения задачи, включающий практический метод, не являющийся гарантированно точным или оптимальным, но достаточный для решения поставленной задачи. Проще говоря, эвристика — это не полностью математически обоснованный (или даже иногда "не совсем корректный"), но при этом практически полезный алгоритм. Его преимущества заключаются в высокой эффективности поиска, простоте реализации и возможности ускоренного решения задачи в тех случаях, когда точное решение не может быть найдено.

Алгоритм планирования движения в любом направлении (алгоритм ПДЛН) основан на алгоритме A* [7—9]. С помощью данного алгоритма по сравнению с алгоритмами A*, D* возможно найти более короткие маршруты движения. Это является следствием того, что в алгоритмах A*, D* движение формируется, главным образом, либо вдоль границ ячейки сетки, либо по ее диагонали. Напротив, алгоритм ПДЛН

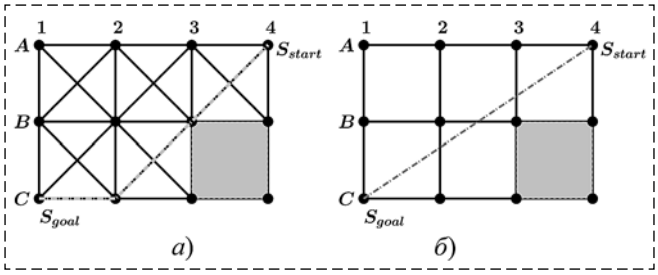

Рис. 1. Примеры траектории движения при фиксированном угле (*а*) и в любом направлении (*б*)

Fig. 1. Geometric interpretation of trajectory planning with fixed angle (*а*) and in any-angle (*б*)

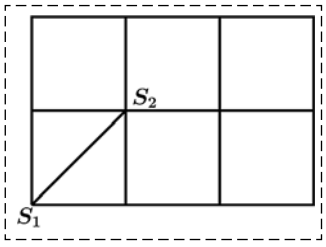

Рис. 2. Вариант 1 обновления пути (Путь 1)

Fig. 2. Option 1 Trajectory updating (Trajectory 1)

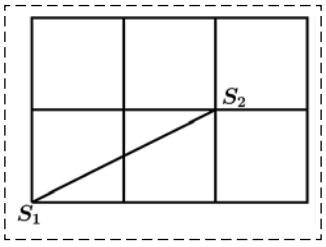

Рис. 3. Вариант 2 обновления пути (Путь 2)

Fig. 3. Option 2 Trajectory updating (Trajectory 2)

(алгоритмы Basic Theta* и Lazy Theta*) не имеет ограничений в выборе направления движения. Например, как видно из рис. 1, *а* и *б*, на траектории, реализованной с помощью алгоритма ПДЛН, удается избежать лишних изменений (отклонений, изгибов) при движении по направлению от начальной точки A4 к целевой точке C1 маршрута. Поэтому с помощью алгоритма ПДЛН возможно в значительной степени сократить результирующий путь. Таким образом, изучение свойств и особенностей данного алгоритма представляет особый интерес.

В алгоритмах ПДЛН траектории движения могут обновляться согласно двум вариантам:

- в первом варианте (рис. 2) путь обновляется между текущей вершиной и соседней вершиной расширения (будем называть его Путь 1);
- во втором варианте (рис. 3) путь обновляется между родительской вершиной (parent) и произвольной вершиной расширения (будем называть его Путь 2).

Описание эвристических алгоритмов поиска маршрута

С помощью эвристических алгоритмов поиска возможно находить маршруты движения от одной вершины (начальной) к другой вершине (целевой, конечной) с наименьшими затратами (стоимостью). Порядок обхода вершин при движении по маршруту определяется эвристической функцией, учитывающей показатели "расстояние + стоимость" и имеющей следующий вид:

$$f(n) = g(n) + h(n), \qquad (1)$$

где $f(n)$ — значение эвристической функции; n — обозначение (номер) рассматриваемой вершины; $g(n)$ — значение функции стоимости достижения рассматриваемой вершины из начальной вершины (может быть как эвристической функцией, так и нет); $h(n)$ — значение функции эвристической оценки расстояния от рассматриваемой вершины до конечной (должна быть допустимой эвристической оценкой).

В общем случае функцию $h(n)$ возможно вычислить различными методами. В данной работе используется один из наиболее распространенных методов оценки расстояния — от рассматриваемой вершины до конечной, а именно:

$$h(n) = \sqrt{(x_t - x_s)^2 + (y_t - y_s)^2}, \qquad (2)$$

где x_t, x_s — значения абсцисс рассматриваемой и целевой вершин; y_t, y_s — значения ординат рассматриваемой и целевой вершин.

Рассмотрим свойства и особенности нескольких алгоритмов поиска маршрутов движения.

Алгоритм A* — это алгоритм поиска, который позволяет найти во взвешенном графе маршрут с наименьшим значением эвристической функции $f(n)$ при движении от начальной вершины до заданной конечной вершины. Алгоритм A* можно упростить так, как показано на рис. 4.

Введем следующие обозначения: s — рассматриваемая вершина; s' — вершина расширения; *parent*(s) — родительская вершина; g(*parent*(s)) — функция стоимости достижения рассматриваемой вершины (*parent*(s)) из начальной вершины; b(*parent*(s), s') — функция стоимости достижения рассматриваемой вершины (s') из вершины *parent*(s); *Neighbour*(s) — соседние вершины, которые можно видеть.

Рассмотрим основные этапы выполнения алгоритма A*.

1. Точка s_0 является начальной вершиной, а также рассматриваемой вершиной. Точки s'_0 (*Neighbour*(s_0)) являются ее соседними верши-

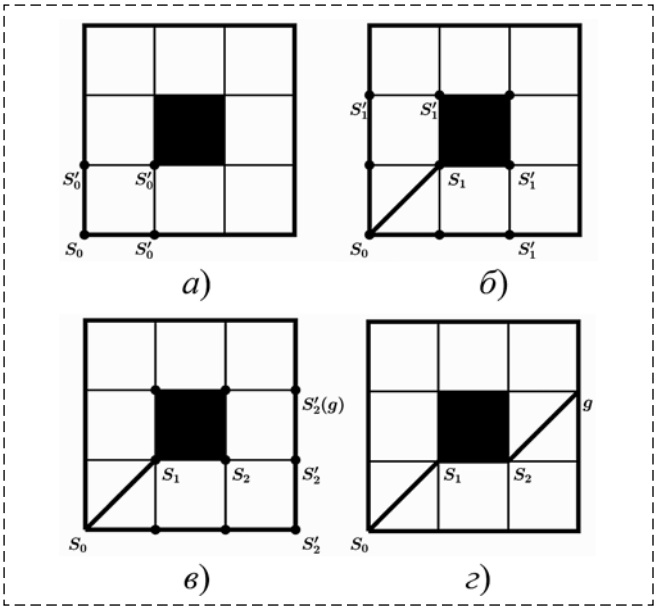

Рис. 4. Упрощенный алгоритм A*
Fig. 4. Brief explanation of Algorithm A*

нами расширения, как показано на рис. 4, *а*. Вычислим все значения эвристической функции $f(s_0)$ для начальной (рассматриваемой) вершины s_0.

2. Выберем вершину расширения s', соответствующую наименьшему значению функции $f(s_0)$. В качестве рассматриваемой вершины принимается вершина s_1 (т. е. $s'_0 = s_1$). Затем обновляются пути так, как показано на рис. 4, *б*. Расширив вершину s_1 в разных направлениях, получим вершины расширения s'_1 (*Neighbour*(s_1) и s'_1, которые не включают вершины расширения на прошлом этапе). Вычислим все значения эвристической функции $f(s_1)$ для рассматриваемой вершины s_1.

3. Выберем вершину расширения s'_1, соответствующую наименьшему значению $f(s_1)$. Далее в качестве рассматриваемой вершины принимается вершина s_2 (т. е. $s'_1 = s_2$). Обновим пути так, как показано на рис. 4, *в*. Расширив вершину s_2, получим вершины расширения s'_2 (*Neighbour*(s_2) и s'_2, которые не включают вершины расширения на прошлом этапе) так, как показано на рис. 4, *в*.

4. Целевая вершина g находится в одной из вершин расширения s'_2. Обновим пути так, как показано на рис. 4, *г*. Таким образом, итоговый путь от начальной до конечной точки маршрута $s_0 - s_1 - s_2 - g$ показан на рис. 4, *г*.

Алгоритм Basic Theta* является модификацией алгоритма A* и имеет определенное сходство с алгоритмом D* [10, 11]. Отличительная особенность алгоритма Basic Theta* — способность обеспечивать движение по короткому пути с меньшим числом изменений в направлении движения. При этом траектория может распространяться в любом направлении, не обязательно через ближайшие вершины, но при условии, что между ними имеется прямая видимость. Основные этапы алгоритма Basic Theta* показаны на рис. 5.

Проанализируем основные этапы алгоритма Basic Theta* (рис. 5).

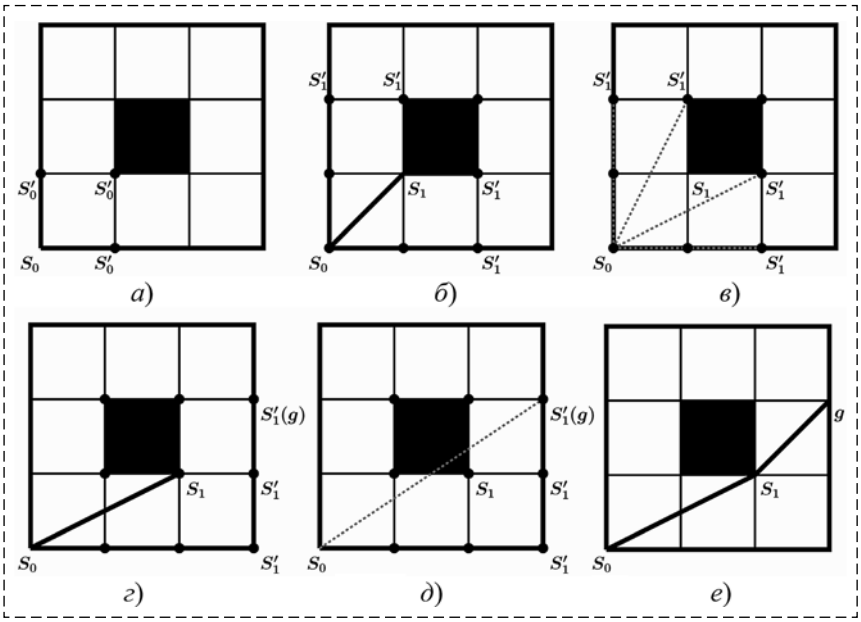

Рис. 5. Упрощенный алгоритм Basic Theta*
Fig. 5. Brief explanation of Algorithm Basic Theta*

1. Содержание этапа аналогично этапу 1 алгоритма A* (см. рис. 4, *а* и 5, *а*).

2. Содержание этапа аналогично этапу 2 алгоритма A* (см. рис. 4, *б* и 5, *б*).

3. Проверка поля зрения (пунктирная линия (рис. 5, *в*) между родительской вершиной (*parent*(s_1) = = s_0) и всеми вершинами расширения s'_1.

Если между родительской вершиной (*parent*(s_1) = s_0) и вершиной расширения s'_1 имеется прямая видимость, то значение эвристической функции $f(s_1) = g(parent(s_1)) + b(parent(s_1), s'_1) + h(s'_1)$.

Если между родительской вершиной (*parent*(s_1) = s_0) и вершиной расширения s'_1 не имеется

прямой видимости, то значение эвристической функции $f(s_1) = g(s_1) + b(s_1, s'_1) + h(s'_1)$.

Выберем вершину расширения s'_1, соответствующую наименьшему значению эвристической функции $f(s_1)$, в качестве рассматриваемой вершины s_1. На рис. 5, *в* показан пример, когда между родительской вершиной и соответствующей вершиной расширения s'_1 имеется прямая видимость.

4. Согласно варианту 2 обновления пути (Путь 2) принимаем ($parent(s_1) = s_0$) и $s'_0 = s_1$. Расширяем вершину s_1 (рис. 5, *г*) и получаем вершины расширения s'_1, которые не включают вершины расширения на предыдущем этапе ($Neighbour(s_1)$ и s'_1).

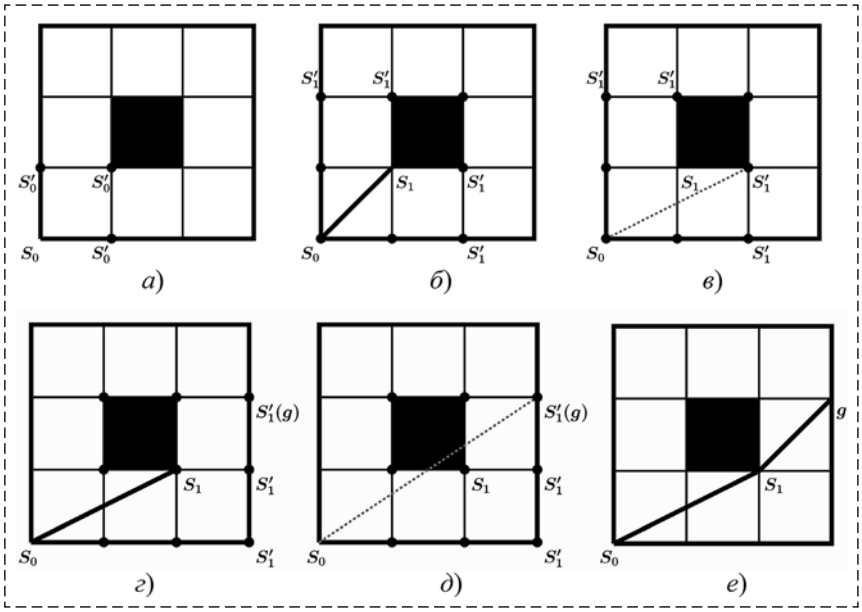

Рис. 6. Упрощенный алгоритм Lazy Theta* и Lazy AT
Fig. 6. Brief explanation of Algorithm Lazy Theta* and Lazy AT

5. Целевая вершина g находится в одной из вершин расширения s'_1. Если между родительской вершиной $parent(s_1) = s_0$ и целевой вершиной g нет прямой видимости (рис. 5, *д*, пунктирная линия), то дальнейшее обновление пути между текущей (рассматриваемой) вершиной s_1 и целевой вершиной g выполняется согласно варианту 1 (Путь 1).

6. На этом работа алгоритма заканчивается. Итоговый путь от родительской вершины до целевой вершины $s_0 - s_1 - g$ показан на рис. 5, *е*.

Алгоритм Lazy Theta* представляет собой модификацию алгоритма Basic Theta* [12], обеспечивающую более высокую скорость вычислений, поскольку в данном алгоритме реализован простой подход к выполнению проверки видимости вершин расширения, находящихся в данный момент в поле зрения из родительской вершины.

Алгоритм Lazy AT — это новый алгоритм, разработанный авторами статьи на основе алгоритма Lazy Theta* в результате оптимизации и комбинирования достоинств предыдущих алгоритмов. Данный алгоритм менее сложен по структуре и обладает более высоким быстродействием. Алгоритм Lazy AT отличается от рассмотренных алгоритмов процедурой поиска рассматриваемой вершины, которая позволяет уменьшить время вычислений. В алгоритме Lazy AT еще добавляется проверка поля зрения между родительской вершиной родителя $parent(parent(s))$ и соответствующей вершиной расширения s', чтобы получить более короткий путь маршрута.

Основные этапы алгоритмов Lazy Theta* и Lazy AT показаны на рис. 6.

Проанализируем основные этапы алгоритмов Lazy Theta* и Lazy AT (рис. 6).

1. Содержание этапа аналогично этапу 1 алгоритма A* (см. рис. 2, *а* и 6, *а*).

2. Содержание этапа аналогично этапу 2 алгоритма A* (см. рис. 2, *б* и 6, *б*).

3. Для алгоритмов Lazy Theta* и Lazy AT (рис. 6, *в*): заранее предполагается, что между родительской вершиной $parent(s_1) = s_0$ и вершиной расширения s'_1 должна быть прямая видимость.

Для алгоритма Lazy Theta*:

▶ Если условие

$$g(parent(s_1)) + b(parent(s_1), s') < g(s'_1)$$

выполняется, то значение эвристической функции

$$f(s_1) = g(parent(s_1)) + b(parent(s_1), s'_1) + h(s'_1).$$

Если условие

$$g(parent(s_1)) + b(parent(s_1), s') < g(s'_1)$$

не выполняется, то значение эвристической функции

$$f(s_1) = g(parent(s_1)) + b(parent(s_1), s'_1) + h(s'_1).$$

Выберем вершину расширения s_1', соответствующую наименьшему значению эвристических функций $f(s_1)$, в качестве рассматриваемой вершины s_1.

Проверяется свойство прямой видимости (рис. 6, *в*) между родительской вершиной ($parent(s_1) = s_0$) и соответствующей вершиной расширения s_1' (проверка выполняется только один раз). На рис. 6, *в* пунктирной линией показано, что между данными вершинами имеется прямая видимость. ◄

Для алгоритма Lazy AT:

► Выберем вершину расширения s_1', соответствующую наименьшему значению эвристических функций

$$f(s_1) = g(parent(s_1) + b(parent(s_1), s_1') + h(s_1'),$$

в качестве рассматриваемой вершины s_1'.

Проверяется свойство прямой видимости между родительской вершиной ($parent(s_1) = s_0$) и соответствующей вершиной расширения s_1' (проверка выполняется только один раз). На рис. 6, *в* пунктирной линией показано, что между данными вершинами имеется прямая видимость.

Затем выполняется проверка свойства прямой видимости между родительской вершиной родителя $parent(parent(s_1))$ и соответствующей вершиной расширения s_1'. Если между родительской вершиной родителя $parent(parent(s_1))$ и соответствующей вершиной расширения s_1' имеется прямая видимость, то дальнейшее обновление пути между родительской вершиной родителя $parent(parent(s_1))$ и соответствующей вершиной расширения s_1' выполняется согласно варианту 2 (Путь 2). ◄

4. Согласно варианту 2 обновления пути (Путь 2) принимаем $parent(s_1) = s_0$ и $s_0' = s_1$. Расширяем вершину s_1' (рис. 6, *г*) и получаем вершины расширения s_1', которые не включают вершины расширения на предыдущем этапе ($Neighbour(s_1)$ и s_1').

5. Целевая вершина g находится в одной из вершин расширения s_1'. Если между родительской вершиной $parent(s_1) = s_0$ и целевой вершиной g нет прямой видимости (рис. 6, *д*, пунктирная линия), то дальнейшее обновление пути между текущей (рассматриваемой) вершиной s_1 и целевой вершиной g выполняется согласно варианту 1 (Путь 1).

Для алгоритма Lazy AT:

► Кроме этого, выполняется проверка условия прямой видимости между родительской вершиной родителя $parent(parent(s_1)) = s_0$ и целевой вершиной g.

Если между родительской вершиной родителя $parent(parent(s_1))$ и целевой вершиной g нет прямой видимости, то дальнейшее обновление пути между родительской вершиной родителя $parent(parent(s_1))$ и целевой вершиной g выполняется согласно варианту 1 (Путь 1). ◄

6. На этом работа алгоритма заканчивается. Итоговый путь от родительской вершины до целевой вершины $s_0 - s_1 - g$ показан на рис. 6, *е*.

Показатели опасности местности, по которой движется марсоход

При нахождении марсохода в какой-либо неизвестной неструктурированной среде (слабо изученной области планеты) особенно важно соблюдать правила безопасности его передвижения.

На основе анализа поверхностных особенностей рельефа планеты можно судить о показателе опасности ее рельефа. Показатель опасности рельефа является одним из оптимизируемых параметров алгоритма.

Анализ показателя опасности рельефа помогает найти для марсохода траекторию с низким уровнем риска.

Топографические особенности

В настоящее время обычно используют колесную конфигурацию марсохода. Основной опасностью при движении по поверхности Марса для колесного марсохода являются наклоны и препятствия. При значении угла наклона поверхности планеты больше его критического значения возможно опрокидывание аппарата. Преодоление препятствий — это одна из основных причин для существенного изменения положения марсохода. Данная ситуация часто возникает на поверхности планеты и может привести к снижению координации между колесами аппарата, а также к его застреванию в ямах, впадинах и других неровностях рельефа.

Прежде чем определить показатели опасности местности, необходимо провести анализ местности.

Исходя из требований устойчивости положения марсохода при его движении и преодо-

лении препятствий были выбраны следующие топографические коэффициенты: угол наклона местности, шероховатость поверхности, размах рельефа.

- ***Анализ угла наклона местности***

Угол наклона местности или крутизна склона (ската) — угол, образуемый направлением склона с горизонтальной плоскостью и выражаемый в угловых мерах или уклонах. Анализ угла наклона местности является одной из важных задач при анализе характеристик местности. Из литературы [13] известно, что в настоящее время наибольшее значение угла наклона местности, при котором марсоход может выполнить подъем, составляет 30°. Поэтому для оценки способности марсохода преодолевать склоны в работе было использовано именно это критическое значение данного показателя.

Введем следующие обозначения: $(i, j, Z_{i,j})$ — координаты положения ячейки на сетке (рис. 7, направление стрелки на сетке указывает направление движения), где i, j — координаты ячейки сетки (индексы); $Z_{i,j}$ — высота ячейки сетки над горизонтальной поверхностью; e — значение шага сетки (размер ячейки).

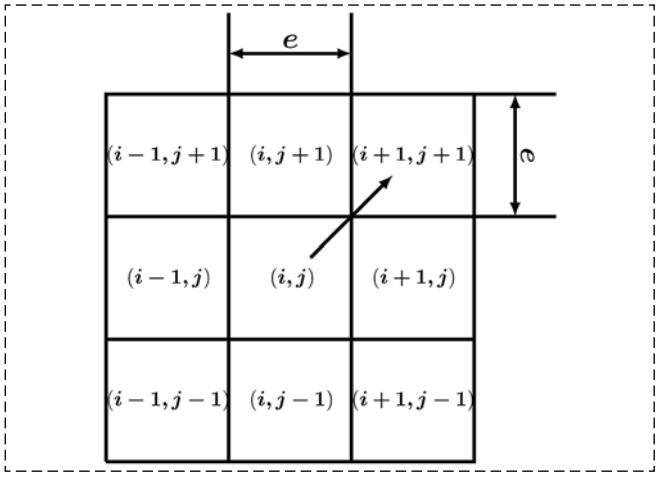

Рис. 7. Определение положения на координатной сетке
Fig. 7. Given position on the coordinate grids

Введем два показателя, характеризующих изменение высоты поверхности:

- в направлении "восток-запад"

$$f_x(i,j) = \frac{Z_{i+1,j+1} + Z_{i+1,j} + Z_{i+1,j-1} - Z_{i-1,j+1} - Z_{i-1,j} - Z_{i-1,j-1}}{6e}; \quad (3)$$

- в направлении "север-юг"

$$f_y(i,j) = \frac{Z_{i-1,j+1} + Z_{i,j+1} + Z_{i+1,j+1} - Z_{i-1,j-1} - Z_{i,j-1} - Z_{i+1,j-1}}{6e}. \quad (4)$$

Тогда угол наклона местности вычисляется по следующему соотношению:

$$\theta(i,j) = \arctan\sqrt{f_x^2(i,j) + f_y^2(i,j)}. \quad (5)$$

- ***Шероховатость поверхности (рельефа)***

Шероховатость поверхности является отражением степени подъема и падений рельефа. Шероховатость может влиять на устойчивость движения марсохода, что важно для обеспечения эффективной работы его оборудования.

Шероховатость рельефа (r) обычно определяется как отношение площади поверхности ($S_1(i, j)$) к его проекции (S) на горизонтальную плоскость, т. е.

$$r(i,j) = S_1(i,j)/S. \quad (6)$$

- ***Размах рельефа***

Размах рельефа — разность абсолютных высот местности или относительное превышение максимальных высот над минимальными. Данный показатель отражает способность марсохода к преодолению препятствий. Из литературы [13] известно, что в настоящее время наибольшая высота препятствия, при которой марсоход может выполнить подъем, составляет 20 см.

Размах рельефа определяется в статье как разность высот двух соседних сеток по направлению движения. Например, на рис. 7 $H(i, j)$ по направлению стрелки вычисляется по следующему соотношению

$$H(i,j) = |\, H(i+1, j+1) - H(i,j). \quad (7)$$

Оценка опасности рельефа

На основе полученных значений топографических коэффициентов возможно охарактеризовать степень опасности местности для движения марсохода.

Показатель опасности, связанный с углом наклона местности, определяется следующим образом:

$$R_1(i,j) = \begin{cases} +\infty \text{ при } \theta > \theta_{th}; \\ 1 + \dfrac{\theta(i,j)^2}{\theta_{th}^{\,2}}, \end{cases} \quad (8)$$

где θ_{th} — максимальный угол наклона местности.

Показатель опасности, связанный с шероховатостью рельефа, определяется следующим выражением:

$$R_2(i,j) = 1 + \frac{r(i,j)}{r_{\max}}, \quad (9)$$

где $r(i, j)$ — шероховатость рельефа; $r_{\max}$ — максимальная шероховатость рельефа.

Показатель опасности, связанный с размахом рельефа, определяется следующим выражением:

$$R_3(i,j) = 1 + \frac{H(i,j)^2}{H_{\max}^2}, \quad (10)$$

где $H(i, j)$ — размах рельефа; $H_{\max}$ — максимальный размах рельефа.

Комплексный показатель опасности местности $R(i, j)$ определяется как максимальная величина одного из трех показателей опасности:

$$R(i,j) = \max(R_1(i,j), R_2(i,j), R_3(i,j)), \quad (11)$$

поскольку наибольший показатель опасности является тем фактором, который сильнее всего влияет на движение марсохода.

Данный комплексный показатель выступает в качестве одного из параметров эвристической функции, которая формируется для нового алгоритма Risk Lazy AT.

Алгоритм Risk Lazy AT

Основываясь на приведенной выше теории, в данной работе предлагается новая модификация алгоритма Lazy AT, а именно алгоритм Risk Lazy AT. Разница между ними заключается в том, что в алгоритме Risk Lazy AT в эвристической функции (см. (1)) учитывается фактор опасности рельефа местности.

Таким образом, новая эвристическая функция приобретает следующий вид:

$$f(n) = g(n) + h(n) + c(n), \quad (12)$$

где $c(n)$ — показатель опасности для маршрута марсохода.

Показатель опасности для маршрута марсохода от точки s_0 до точки s_m определяется следующим образом:

$$c(s_0, s_m) = \sum_{(i,j)=s_0}^{(i,j)=s_m} \alpha R(i,j), \quad (13)$$

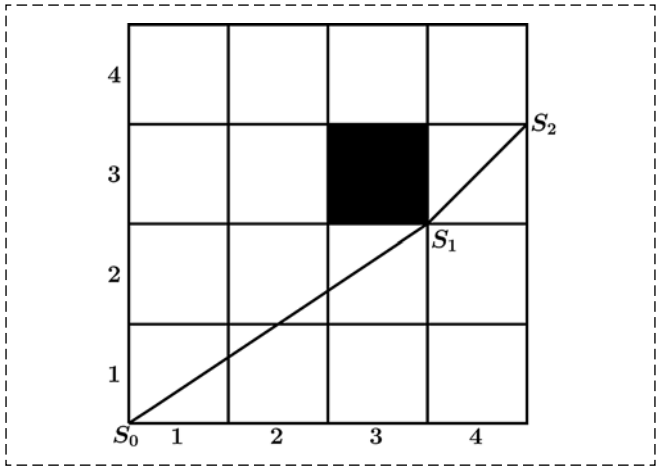

Рис. 8. Пример определения показателя опасности для маршрута

Fig. 8. Explanation of the definition of the hazard indicator for trajectory

где s_0 — начальная точка маршрута; s_m — конечная точка маршрута; $R(i, j)$ — показатель опасности рельефа местности на координатной сетке (i, j); α — соответствующие весовые коэффициенты.

Если длина пути через ячейку сетки меньше $\sqrt{2}/2e$, весовой показатель $\alpha = 0{,}5$. Если длина пути через сетку больше $\sqrt{2}/2e$, весовой показатель $\alpha = 1$. Здесь e — значение шага сетки. Например, для маршрута, показанного на рис. 8, показатель опасности имеет следующий вид:

$$c(s_0, s_2) = R(1,1) + 1/2(R(2,1) + R(2,2)) + R(3,2) + R(4,3). \quad (14)$$

Результаты моделирования

На первом этапе был проведен сравнительный анализ следующих четырех алгоритмов: A*, Basic Theta*, Lazy Theta* и Lazy AT. Результаты моделирования алгоритмов для размеров области на координатной сетке 50×50 м приведены в табл. 1, а для размеров области 100×100 м — в табл. 2. Моделирование проводили при следующих условиях: шаг координатной сетки равен 1 м; число случайных препятствий в области — 10 %, 20 %, 30 %; L, м — длина пути; t, с — время вычисления.

Результаты моделирования показывают, что путь, полученный с помощью алгоритма Lazy AT, является самым коротким из всех. Хотя время вычисления для данного алгоритма больше, чем для алгоритма A*, однако по сравнению со временем вычисления с помощью

Таблица 1

Размер области на координатной сетке 50×50 м

Число случайных препятствий	A*		Basic Theta*		Lazy Theta*		Lazy AT	
	L, м	*t*, с	*L*, м	*t*, с	*L*, м	*t*, с	*L*, м	*t*, с
10 %	70,468	0,243	70,225	0,335	70,225	0,274	69,937	0,269
20 %	72,811	0,223	72,277	0,277	71,998	0,263	70,691	0,244
30 %	75,397	0,214	76,176	0,249	74,212	0,242	72,790	0,225

Таблица 2

Размер области на координатной сетке 100×100 м

Число случайных препятствий	A*		Basic Theta*		Lazy Theta*		Lazy AT	
	L, м	*t*, с	*L*, м	*t*, с	*L*, м	*t*, с	*L*, м	*t*, с
10 %	142,350	0,480	142,350	0,608	142,350	0,532	140,757	0,518
20 %	149,279	0,468	146,688	0,565	160,305	0,524	145,025	0,506
30 %	153,622	0,440	151,318	0,512	150,590	0,503	148,347	0,469

алгоритмов планирования движения в любом направлении (Basic Theta*, Lazy Theta*) оно оказывается меньше.

При увеличении числа препятствий время работы и путь алгоритма Lazy AT заметно меньше, чем у остальных алгоритмов ПДЛН, что еще раз доказывает эффективность данного алгоритма.

На втором этапе моделирования был проведен сравнительный анализ алгоритмов Lazy Theta* и разработанного алгоритма Risk Lazy AT.

Для моделирования данных алгоритмов в качестве примера была создана трехмерная карта произвольной области поверхности планеты размером 30×30 м. Для построения маршрута без столкновения с препятствиями необходимо увеличить размеры препятствий. Результаты работы алгоритмов Lazy Theta* и Risk Lazy AT с учетом препятствий, нанесенных на координатную сетку, показаны на рис. 9 (см. четвертую сторону обложки).

На рис. 9 обозначено: черный кружок — реальные препятствия; красные кружки — границы виртуальных препятствий (увеличение размеров препятствий необходимо для их гарантированного обхода и предотвращения столкновения); синий кружок — расширенная определенная точка (*s*); зеленая непрерывная линия — путь, построенный с помощью алгоритма Lazy Theta*; красная непрерывная линия — путь, построенный с помощью алгоритма Risk Lazy AT.

Результаты моделирования алгоритмов для размеров области на координатной сетке 30×30 м приведены в табл. 3 (здесь *c* — показатель опасности маршрута).

Таблица 3

Число препятствий	Risk Lazy AT		Lazy Theta*	
	L, м	*c*	*L*, м	*c*
2 %	43,147	1,213	43,401	3,378

Анализ полученных результатов показывает, что показатель опасности маршрута (*c*), построенного с помощью алгоритма Risk Lazy AT, меньше чем аналогичный показатель, полученный с помощью алгоритма Lazy Theta*. Следовательно, первый маршрут безопаснее второго.

На третьем этапе моделирования исследовали применение алгоритма Risk Lazy AT для планирования маршрута движения марсохода по поверхности Марса. На основе реальных данных о рельефе поверхности Марса, полученных из материалов USGS [14], была создана трехмерная карта участка поверхности планеты размером 330×330 м (рис. 10, *а*, см. четвертую сторону обложки) с размером пикселя 0,5 м.

Выберем на топографической карте поверхности Марса участок размером 30×30 м (см. на рис. 10, *а* участок, выделенный розовым цветом). Его укрупненный вид показан на рис. 10, *б*, см. четвертую сторону обложки).

На последнем этапе моделирования с помощью алгоритма Risk Lazy AT были получены результаты планирования пространственного маршрута движения марсохода по поверхности

планеты с учетом рельефа местности, наличия опасностей и препятствий (рис. 11, см. четвертую сторону обложки).

Заключение

Проведен сравнительный анализ известных алгоритмов планирования траектории движения подвижного объекта: A*, Basic Theta*, Lazy Theta*. На основе идеи уменьшения сложности алгоритма и протяженности маршрута предложен новый алгоритм планирования траектории движения подвижного объекта Lazy AT.

На основе особенностей топографии поверхности Марса введен показатель опасности, который включен в качестве эвристического параметра в модификацию нового алгоритма планирования траектории движения Risk Lazy AT.

Проведено моделирование алгоритмов A*, Basic Theta*, Lazy Theta*, Lazy AT и выполнен сравнительный анализ полученных результатов по следующим показателям: L — длина пути; t — время вычисления. Результаты моделирования показали, что алгоритм Lazy AT обеспечивает движение подвижного объекта между исходной и конечной точкой маршрута по кратчайшему пути. По сравнению с алгоритмами Basic Theta* и Lazy Theta* время вычислений с помощью алгоритма Lazy AT также меньше.

Сравнительный анализ результатов моделирования алгоритмов Risk Lazy AT и Lazy Theta* показал, что предложенный новый алгоритм Risk Lazy AT может обеспечить безопасное движение подвижного объекта по поверхности планеты.

На основе реальных данных о рельефе поверхности Марса была разработана цифровая карта поверхности планеты и проведено моделирование пространственного маршрута движения марсохода с помощью нового алгоритма Risk Lazy AT. Полученные результаты подтвердили эффективность предложенного алгоритма для планирования безопасного маршрута движения марсохода по пространственной траектории с учетом рельефа поверхности Марса.

Список литературы

1. **Carsten J., Rankin A., Ferguson D., Stentz A.** Global path planning on board the mars exploration rovers // Aerospace Conference. 2007. С. 1—11.

2. **Тань Л., Фомичев А.** Планирование пространственного маршрута полета беспилотного летательного аппарата с использованием методов частично целочисленного линейного программирования // Вестник Московского государственного технического университета им. Н. Э. Баумана. Сер. "Приборостроение". 2016. № 2 (107).

3. **Ferguson D., Stentz A.** Using interpolation to improve path planning: The Field D* algorithm // Journal of Field Robotics. 2006. Vol. 23, N. 2. P. 79—101.

4. **Фомичев А. В., Тань Л.** Разработка алгоритма быстрой компенсации погрешностей комплексированной инерциально-спутниковой системы навигации малогабаритных беспилотных летательных аппаратов в условиях сложной среды // Наука и образование: научное издание МГТУ им. Н. Э. Баумана. 2015. № 10.

5. **Фомичев А. В., Ван Г.** Сравнительный анализ методов планирования маршрута движения для планетохода // Информационные системы и технологии. 2016.

6. **URL:** https://ru.wikipedia.org/wiki/Эвристический_алгоритм (дата обращения 25.04.2018).

7. **Nash A., Koenig S.** Any-angle path planning // AI Magazine. 2013. Vol. 34, N. 4. P. 85—107.

8. **Choi S., Lee J. Y., Yu W.** Fast any-angle path planning on grid maps with non-collision pruning // Robotics and Biomimetics (ROBIO), 2010 IEEE International Conference. 2010. P. 1051—1056.

9. **Uras T., Koenig S.** Speeding-Up Any-Angle Path-Planning on Grids // ICAPS. 2015. P. 234—238.

10. **Nash A., Daniel K., Koenig S., Aelner A.** Theta*: any-angle path planning on grids // AAAI. 2007. Vol. 7. P. 1177—1183.

11. **Mendonca P., Goodwin S.** C-Theta*: Cluster Based Path-Planning on Grids // Computational Science and Computational Intelligence (CSCI), 2015 International Conference. 2015. P. 605—608.

12. **Nash A., Koenig S. and Tovey C.** Lazy theta*: Any-angle path planning and path length analysis in 3d //Third Annual ymposium on Combinatorial Search. 2010.

13. **URL:** https://en.wikipedia.org/wiki/Spirit_(rover) (дата обращения 25.04.2018).

14. **URL:** https://astrogeology.usgs.gov/search/map/Phobos/MarsExpress/HRSC/Phobos_ME_HRSC_DEM_Global_2ppd (дата обращения 25.04.2018).

Development of a Non-hazardous Path Planning Algorithm for Mars Rover in Real Terrain Enviroment

G. Wang, guoyan@yandex.ru, **A. V. Fomichev,** a.v.fomichev@bmstu.ru,
Bauman Moscow State Technical University, Moscow, 105005, Russian Federation

Corresponding author: **Fomichev Aleksei V.,** Ph. D., Associate Professor,
Bauman Moscow State Technical University, Moscow, 105005, Russian Federation,
e-mail: a.v.fomichev@bmstu.ru

Accepted July 30, 2018

Abstract

In order to fulfill the corresponding task successfully, a crucial issue should be addressed is the path planning for the exploration of the Mars surface owing to the environmental features of the tough terrain. Traditional path planning algorithms, such as the A algorithm and the improved A* algorithm — the algorithm D* and the Field D*, which have been successfully implemented on the planetary rover during the expeditions of the Moon and Mars, have the problem of finding the shortest optimal path. One of the more effective algorithms derived from the modified A* refers to the Basic Theta* or the Lazy Theta* algorithms, which are faster any-angle path planning. Additionally, the algorithms can find shorter routes. In this paper, derived from a comprehensive comparison of the existing algorithms (A*, Basic Theta* and Lazy Theta*), a novel modification of the Lazy AT methodology is proposed to reduce the calculation time and obtain a shorter path. Based on the analysis of the surface feature of the Mars topography, the corresponding safety indicator is discussed. The principal hazards of the wheeled vehicles during the exploration on the surface of the Mars are the slopes and the obstacles. According to the requirements for avoiding obstacles as well as the exploration stability of the Mars rover in the period of the exploration, the following topographic coefficients have been chosen to develop the hazard indicator, i.e., the inclination angle of the terrain, the surface roughness and the height difference of the terrain. In addition, to obtain a safe trajectory in algorithm Lazy AT on the Mars surface, the terrain hazard indicator (risk indicator) for the modification of the Risk Lazy AT algorithm is also proposed in this paper. The comparing analysis modeling results of the Risk Lazy AT and Lazy Theta* has shown that our proposed algorithm Risk Lazy AT can guarantee the safety movement of a mobile object during the exploration on the surface of the planet. In light of the real-world surface features of the Mars terrain, the digital map of the planet's surface has been developed and the spatial routing of the rover has been tested with our novel proposed algorithm, so-called Risk Lazy AT.*

Keywords: *algorithm A*, Any-Angle Path Planning, algorithm Basic theta*, algorithm Lazy Theta*, algorithm Lazy AT, algorithm Risk Lazy AT, terrain hazard indicator*

For citation:

Wang G., Fomichev A. V. Development of a Non-hazardous Path Planning Algorithm for Mars Rover in Real Terrain Enviroment, *Mekhatronika, Avtomatizatsiya, Upravlenie,* 2018, vol. 19, no. 11, pp. 734—743.

DOI: 10.17587/mau.19.734-743

References

1. **Carsten J., Rankin A., Ferguson D., Stentz A.** Global path planning on board the mars exploration rovers. In Aerospace Conference, 2007 IEEE (pp. 1—11).

2. **Tan Liguo, Fomichev A. V.** *Planirovanie prostranstvennogo marshruta poleta bespilotnogo letatelqnogo appalata s ispolqzovaniem metodov chactichno celochislennogo linejnogo programmirovaniya* (The spatial flight route planning of unmanned aerial vehicles using the methods of mixed-integer linear programming), *Vectnik moskovckogo gosudarstvennogo technicheckogo universiteta im. N. E. Baumana. Seriya "Priborostroenie"*, 2016 (in Russian).

3. **Ferguson D., Stentz A.** Using interpolation to improve path planning: The Field D* algorithm, *Journal of Field Robotics*, 23.2 (2006): 79—101.

4. **Fomichev A. V., Tan Liguo.** *Razrabotka algoritma byhstroj kompensacii pogreshnostej kompleksirovannoj inercialqno-sputnikoj sistemyh navigacii malogabaritnyhch bespilotnyhch letatelqnyhch apparatov v usloviyach slozhnoj sredyh* (Development of an algorithm for rapid error-compensation for small unmanned aerial vehicles with a complex inertial satellite navigation system in a complex environment), *Nauka i obrazovanie: nauchnoe izdanie MGTU im. N. E. Baumana*, 2015, no. 10 (in Russian).

5. **Fomichev A. V., Wang Guoyan.** *Sravnitelqnyhq analiz metodov planirovaniya marshruta dvizheniya dlya planetochoda* (Comparison analysis of path-planning methodologies for planetary rover), *Informacionnyhe Sistemyh i Technologii*, 2016 (in Russian).

6. **Available at:** https://ru.wikipedia.org/wiki/Эвристический_алгоритм/ (date of access 25.04.2018).

7. **Nash A., Koenig S.** Any-angle path planning, *AI Magazine,* 2013, vol.34, no. 4., pp. 85—107.

8. **Choi S., Lee J. Y., Yu W.** Fast any-angle path planning on grid maps with non-collision pruning. Robotics and Biomimetics (ROBIO), *2010 IEEE International Conference on. IEEE*, 2010.

9. **Uras T., Koenig S.** Speeding-Up Any-Angle Path-Planning on Grids, *ICAPS*, 2015, pp. 234—238.

10. **Nash A., Daniel K., Koenig S.** Aelner A. Theta*: Any angle path planning on grids, *Proceedings AAAI Conference on Artificial Intelligence*, 2007, vol. 7, pp. 1177—1183.

11. **Mendonca P., Goodwin S.** C-Theta*: Cluster based Path-Planning on Grids. International Conference on Computational Science and Computational Intelligence.2015. Mendonca, Pramod, and Scott Goodwin. "C-Theta*: Cluster Based Path-Planning on Grids." Computational Science and Computational Intelligence (CSCI), *2015 International Conference on. IEEE*, 2015, pp. 605—608.

12. **Nash A., Koenig S., Tovey C.** Lazy theta*: Any-angle path planning and path length analysis in 3d, *Third Annual Symposium on Combinatorial Search,* 2010.

13. **Available at:** https://en.wikipedia.org/wiki/Spirit_ (rover) / (date of access 25.04.2018).

14. **Available at:** https://astrogeology.usgs.gov/search/map/Phobos/MarsExpress/HRSC/Phobos_ME_HRSC_DEM_Global_2ppd / (date of access 25.04.2018).