

УДК 621.865.8

*М. Н. МИРОНОВА*, канд. техн. наук, доц.

**В. В. ЦАРАНЮК**

Белорусско-Российский университет (Могилев, Беларусь)

## **АЛГОРИТМЫ УПРАВЛЕНИЯ ДВИЖЕНИЕМ МОБИЛЬНОГО РОБОТА**

### **Аннотация**

Рассмотрены вопросы управления движением мобильного робота. Приведены алгоритмы планирования траектории его перемещения.

### **Ключевые слова:**

мобильный робот, планирование траектории, управление.

В настоящее время мобильные роботы получили широкое практическое применение во многих сферах человеческой деятельности. При этом основной проблемой мобильных роботов является решение задачи навигации. Общепринятый подход к навигации робота заключается в составлении карт рабочего пространства и локализации, информация о которой необходима для решения большинства задач управления: прохождение заданной траектории, поиск траектории к заданной точке, возвращение в исходное положение.

Однако многие роботизированные задачи могут быть решены без какой-либо карты, используя подход пассивной навигации. Например, перемещаться по лабиринту, следуя стене. Очень простой класс роботов, достигающих цели, известен как тележка Брайтенберга и характеризуется прямой связью между датчиками и двигателями. У них нет четкого внутреннего представления об окружающей среде, в которой они работают. Чтобы двигаться по определенному вектору, нужно оценить направление вектора в текущем месте, а для этого требуется как минимум два измерения положения. Для этого можно использовать два датчика двустороннего обнаружения, по одному на каждой стороне тела робота.

Другой класс реактивных роботов известен как «bugs» – простые автоматы, которые выполняют поиск цели при наличии непроходимых областей или препятствий. Существует большое количество алгоритмов для таких роботов, обладающие способностью определять, когда они находятся в непосредственной близости от препятствия. В этом отношении они похожи на тележку Брайтенберга, но bug включают в себя конечный автомат и другую логику между датчиком и двигателями. У этих роботов есть память, которой не было у машины Брайтенберга. Разработано множество вариантов bug-алгоритма, но, повышая производительность в одном типе среды, они могут снижать производительность в других. Принципиально робот ограничен тем, что не использует карту. Он не может видеть общую картину и поэтому выбирает пути, оптимальные локально, а не глобально.

Известно, что средством к достижению наилучшего пути между двумя

точками является использование карты окружающей среды. Как правило, «лучшее» означает кратчайшее расстояние, но оно также может включать в себя некоторые затраты, связанные с проходимостью, т. е. насколько легко проехать по местности. Более сложный планировщик может также учитывать размер робота, кинематику и динамику транспортного средства и избегать путей, которые включают повороты, более крутые, чем может выполнить транспортное средство.

Существует множество способов представить карту и положение робота на карте. Для представления мест и путей между ними могут использоваться графы. Они могут быть эффективно проанализированы, чтобы найти путь, который минимизирует некоторые затраты, чаще всего пройденное расстояние. Более простым и удобным для компьютера представлением является сетка занятости, которая широко используется в робототехнике. Сетка занятости рассматривает мир как сетку ячеек, где каждая ячейка помечается как занятая или незанятая.

Популярным алгоритмом планирования пути робота является  $D^*$  (алгоритм инкрементного поиска), который находит наилучший путь через граф, который он сначала вычисляет и который соответствует входной сетке занятости. Алгоритм  $D^*$  имеет ряд функций, полезных для реальных приложений.

Во-первых, он обобщает сетку занятости до карты стоимости, которая представляет стоимость прохождения каждой ячейки в горизонтальном или вертикальном направлении.  $D^*$  находит путь, который минимизирует общую стоимость пути. Если интересует кратчайшее время для достижения цели, то стоимость – это время, затрачиваемое на проход через ячейку, и она обратно пропорциональна проходимостью. Если заинтересованы в минимизации ущерба транспортному средству или максимальном комфорте для пассажиров, тогда стоимость может быть связана с неровностями местности внутри ячейки. Затраты, назначенные ячейкам, также будут зависеть от характеристик робота: большой четырехколесный робот может иметь конечную стоимость пересечения местности, тогда как для маленького робота эта стоимость может быть бесконечной.

Во-вторых, алгоритм  $D^*$  поддерживает поэтапное перепланирование. Это важно, если во время движения обнаруживается, что пространство отличается от карты. Если мы обнаружим, что маршрут имеет более высокую, чем ожидалось, стоимость или полностью заблокирован, мы можем поэтапно перепланировать его, чтобы найти лучший путь. Инкрементное перепланирование требует меньших вычислительных затрат, чем полное перепланирование.

Алгоритм  $D^*$  позволяет обновлять карту в любое время во время движения робота. После перепланировки робот просто перемещается в соседнюю ячейку с наименьшей стоимостью, что обеспечивает непрерывность движения даже при изменении плана.

Преимущество алгоритма  $D^*$  заключается в возможности эффективно изменять карту стоимости во время движения. Это довольно распространенное требование в робототехнике, поскольку реальные датчики имеют конечный диапазон, и робот открывает больше пространства по мере продвижения (рис. 1).

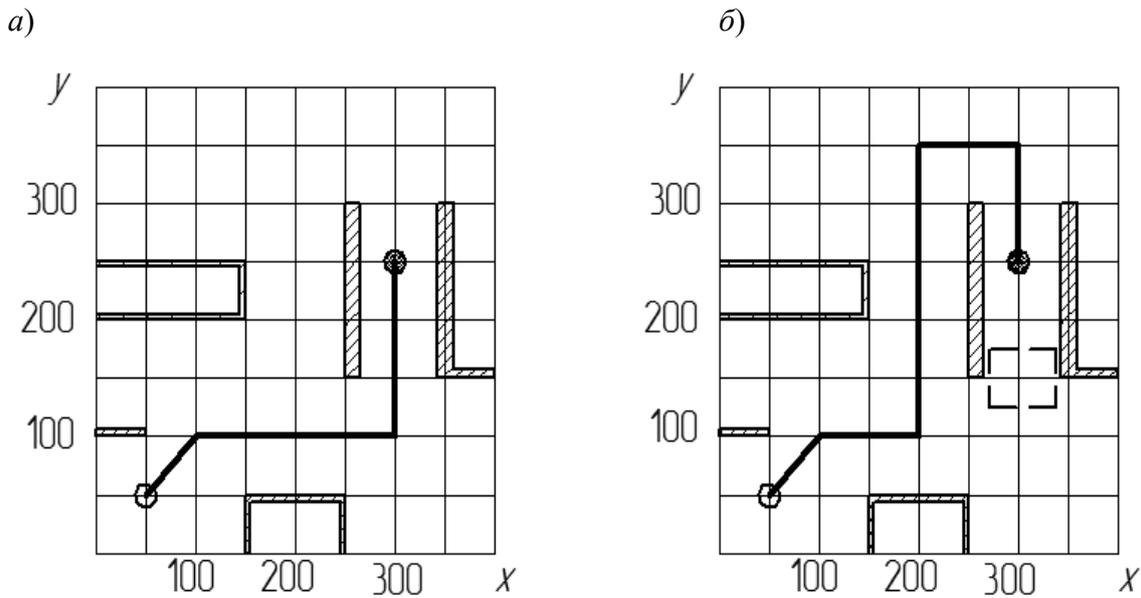


Рис. 1. Пути из планировщика  $D^*$ : *a* – с исходной картой; *б* – с измененной картой (область с более высокой стоимостью обозначена пунктирным прямоугольником)

Известно, что планирование пути робота осуществляется в два этапа. На этапе планирования осуществляется анализ карты. Фаза запроса используется для нахождения кратчайшего пути. Алгоритмы планирования, такие как дистанционное преобразование и  $D^*$ , требуют значительного объема вычислений, но фаза запроса очень малозатратна. Однако план всегда зависит от цели. И если цель меняется, то этап планирования должен быть выполнен повторно. Несмотря на то, что алгоритм  $D^*$  позволяет пересчитывать путь при изменении карты затрат, он не поддерживает изменение цели.

Несоответствие затрат на планирование и запросы привело к разработке дорожной карты, метода, в котором запрос может включать как начальную, так и конечную позиции. Фаза планирования обеспечивает анализ, который поддерживает изменение отправных точек и изменение целей.

Тогда проблема навигации роботов становится задачей построения сети свободных от препятствий путей через окружающую среду. В литературе такая сеть называется дорожной картой. Дорожная карта должна быть рассчитана только один раз и затем ее можно использовать, чтобы перемещаться из любого начального местоположения в любое целевое местоположение.

Первый шаг создания дорожной карты – найти свободное место на карте, которое является дополнением к занятому пространству и представляет собой матрицу с ненулевыми элементами, по которой робот может свободно перемещаться. Граница также является препятствием, поэтому крайние ячейки помечаются как несвободные. На втором этапе вычисляется топологический скелет свободного пространства морфологическим образом. Далее к свободному пространству применяется алгоритм обработки, известный как прореживание.

Высокая вычислительная стоимость рассмотренного метода создания дорожных карт делает их неприменимыми для больших карт и привела к

развитию вероятностных методов. Наиболее известным из этих методов является вероятностная дорожная карта или метод PRM.

При методе PRM на этапе планирования находятся  $N$  случайных точек, которые лежат в свободном пространстве. Каждая точка соединена со своими ближайшими соседями прямым путем, не пересекающим никаких препятствий, чтобы создать сеть или граф с минимальным количеством непересекающихся компонентов и без циклов. Преимущество PRM заключается в том, что необходимо проверить относительно небольшое количество точек, чтобы убедиться, что точки и пути между ними свободны от препятствий. Примеры графиков, созданных при планировании методом PRM, показаны на рис. 2. Точки представляют собой случайно выбранные места, а линии – пути без препятствий.

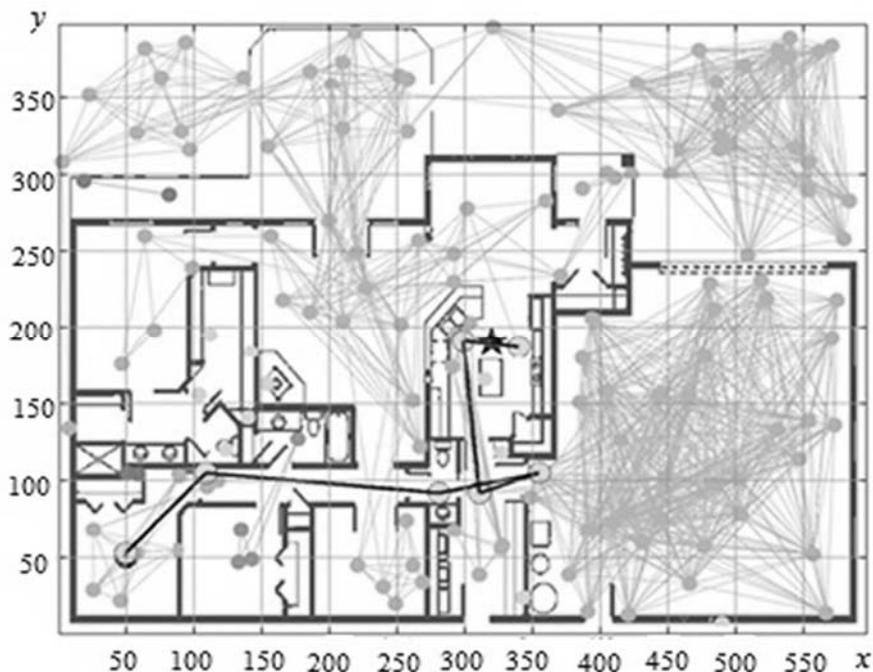


Рис. 2. Планировщик вероятностных дорожных карт (PRM) и графики, созданные на этапе планирования

Для фазы запроса, формирующей путь от начальной точки к цели с минимальной стоимостью, в методе PRM предоставляется начальная и конечная позиция робота. После создания дорожной карты на этапе планирования можно изменить цель и отправные точки, при этом нужно только повторить этап запроса.

Однако следует учитывать, что каждый раз при запуске планировщика создается другая дорожная карта, что приводит к различным путям и длинам. Планировщик может потерпеть неудачу, создав сеть, состоящую из непересекающихся составных частей. В то же время длинные узкие промежутки между препятствиями вряд ли будут использоваться, поскольку вероятность случайного выбора точек, лежащих вдоль таких пространств, очень мала.

Рассмотренные ранее планировщики генерировали пути, независимые от движения, которые робот действительно может достичь. Альтернативой является проектирование пути с самого начала, по которому может следовать мобильный робот.

Как правило, робот перемещается между дискретными точками в пространстве. Первоначально робот находится в исходной точке и может двигаться вперед к трем точкам, показанным на рис. 3, *а*. В конце каждой ветви пути добавляется один и тот же набор из трех движений, соответствующим образом повернутых и перемещенных (рис. 3, *б*). В результате получается граф, содержащий большое количество узлов и путей. За счет увеличения количества итераций можно заполнить больше возможных путей.

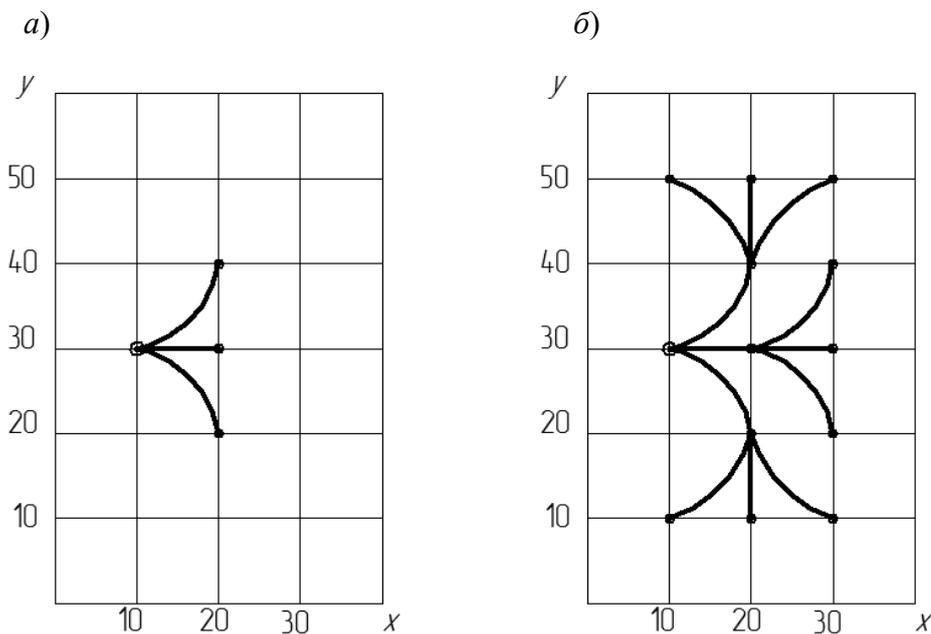


Рис. 3. План решетки после одной и двух итераций

После создания решетки можно вычислить оптимальный путь между любыми двумя узлами, используя метод запроса.

Если стоимость перемещения по трем вариантам управления (прямо, повернуть налево и повернуть направо) равна, то можно получить путь, показанный на рис. 4, *а*. А если стоимость перемещения по дуге увеличить, то получится путь, показанный на рис. 4, *б*, который имеет меньше поворотов. Однако путь длиннее – пять, а не четыре сегмента.

Пути, сгенерированные планировщиком простых решеток, могут управляться роботом, но возникают проблемы с движением по диагонали. При наличии препятствий в окружающей среде планировщик будет генерировать непрерывную последовательность резких левых и правых поворотов, которых можно избежать, используя более усовершенствованную версию планировщика решеток, основанную на построении пути с помощью сотни дуг вместо трех.

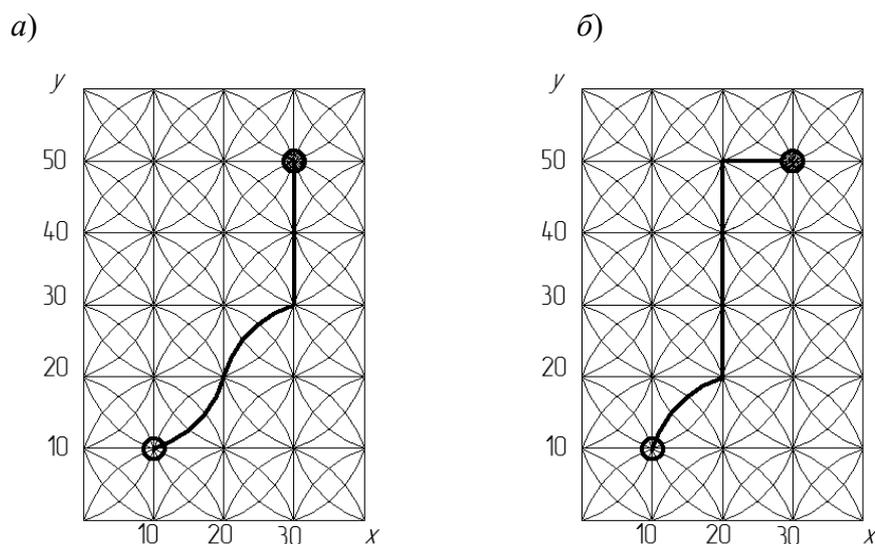


Рис. 4. Пути на решетчатом графе: *a* – с единой стоимостью; *б* – с повышенными затратами за повороты

Каждый из рассмотренных выше методов управления имеет как достоинства, так и недостатки. Некоторые методы плохо подходят для систем реального времени, т. к. имеют низкое быстродействие. Иные методы имеют низкую точность управления, либо сложны в реализации. Некоторые методы сложны для понимания человеком.

На основе изучения методик проектирования траекторий перемещения роботов планируется разработка методов и алгоритмов для управления движением мобильного робота Robotino.

Контакты:

MarinaMN16@mail.ru (Миронова Марина Николаевна);

vera.tsaranok@mail.ru (Царанок Вероника Витальевна).

***M. N. MIRONOVA, V. V. TSARANOK***

## **ALGORITHMS FOR CONTROLLING THE MOVEMENT OF A MOBILE ROBOT**

### **Abstract**

The issues of motion control of a mobile robot are considered. Algorithms for planning the trajectory of its movement are given.

### **Keywords:**

mobile robot, trajectory planning, control.