

АДАПТАЦИЯ И МОДЕЛИРОВАНИЕ АЛГОРИТМА ПОСТРОЕНИЯ МАРШРУТА ДЛЯ БЕСПИЛОТНОГО ЛЕТАТЕЛЬНОГО АППАРАТА С ИСПОЛЬЗОВАНИЕМ СРЕДЫ СИМУЛЯЦИИ V-REP

Домницкий Е.А. (Федеральное государственное автономное образовательное учреждение высшего образования «Национальный исследовательский университет ИТМО»),
Научный руководитель – кандидат технических наук, доцент Вискнин И.И.
(Федеральное государственное автономное образовательное учреждение высшего образования «Национальный исследовательский университет ИТМО»)

Аннотация

В настоящем докладе предложено использовать алгоритм A* построения маршрута для БПЛА в пространстве, а также простейшие условия триггера на перестроение маршрута в случае встречи со случайными препятствиями на пути. Выбор алгоритма соответствует набору определенных в докладе требований. Решение смоделировано при помощи среды симуляции V-REP.

Введение

Задача навигации и ориентации в пространстве, достижения конечной цели передвижения, обхождения препятствий является одной из ключевых при разработке автономных робототехнических систем как наземных, так и летательных, действующих на местности.

На данный момент существуют различные подходы к решению данной проблемы, во многом они зависят от условий. Можно найти простейшие алгоритмы по типу BUG и CBUG, являющиеся по сути не алгоритмами построения маршрута, но алгоритмами избегания препятствий на пути к цели. Эти алгоритмы работают на плоскости и применяются в-основном для наземных роботов. Чаще всего задача поиска оптимального маршрута решается при помощи представления пространства в качестве графа, в котором отмечается стартовый узел и узел цели, кратчайший путь до которой надо найти. Очевидно, что пути в данном случае находят при помощи алгоритмов по графам: при помощи алгоритма волновой трассировки, или алгоритма Беллмана-Форда, Джонсона, Дейкстры и т.п.. Узлы, в которых находятся препятствия, помечаются как непроходимые и рёбра до них удаляются из графа. Также существуют мета-эвристические мульти-агентные подходы, например алгоритм *Ant Colony Optimization Based On Population*. Данный алгоритм заключается в поиске оптимального маршрута путём многократного исследования пространства поиска в различных направлениях.

Однако, конкретный случай – случай задачи для беспилотных летательных аппаратов – может накладывать ограничения и выдвигать определенные требования. В данной работе БПЛА будет рассматриваться как агент децентрализованной системы (роя или коллектива), который при этом должен обладать достаточной степенью независимости от коллектива или роя, чтобы иметь возможность функционировать автономно от других агентов в некоторых ситуациях.

Информация о маршруте и его перестроениях в таком случае должна быть представлена в элементарном виде для простоты передачи ее другим агентам или использования информации о маршруте в проверках на наличие скрытых деструктивных информационных воздействий. Самый простой и универсальный вариант – маршрут представляется как последовательность точек в пространстве. Также алгоритм поиска маршрута должен быть полным в том смысле, что он всегда должен находить решение, если таковое существует. Помимо этого, искомый путь должен быть оптимальным, что является NP-сложной задачей.

В связи со всеми вышеперечисленными требованиями, наиболее подходящим видится применение эвристических алгоритмов. Автором данного доклада был выбран для реализации алгоритм A* с эвристикой Манхеттена.

Дизайн эксперимента

Экспериментальная сцена симуляции сделана с использованием среды симуляции V-REP, объекты внутри сцены управляются посредством связанных с ними скриптов в Regular API среды на языке Lua. Сцена использует встроенный в среду симуляции физический движок Bullet 2.78, шаг симуляции стандартный – 50 мс. В центре сцены расположена поверхность 15x15 м «пол» симуляции. Точка отсчета координат расположена в центре поверхности.

Для представления пространства сцены выбран метод Discretize Space (дискретное пространство). Таким образом, создается навигационная кубическая сетка ограниченных размеров: 160x160x84 узла – от -80 до 80 по оси X, от -80 до 80 по оси Y, от -4 до 80 по оси Z. Расстояние между двумя смежными узлами сетки, находящимися на ребре «куба», равно 0.1 м в сцене симуляции. Иными словами, координатам (-34, 57, 20) в навигационной сетке соответствуют координаты (-3.4, 5.7, 2.0) в симуляции. Переход к целым числам обусловлен особенностями хранения навигационной сетки в виде таблиц и их индексации в Lua.

В качестве БПЛА используется предустановленная в V-REP модель квадрокоптера, приблизительными размерами 0.35x0.35x0.02 м. Управление ей осуществляется посредством перемещения в пространстве зелёной прозрачной сферы – таргета. Сам же корпус коптера перемещается путем перераспределения мощности между пропеллерами в зависимости от взаимного расположения относительно друг друга сферы и корпуса, следуя за сферой (по принципу инверсной кинематики).

Для обнаружения препятствий и их отметки на навигационной сетке используется настраиваемая предустановленная в симулятор модель лидара. Его корпус имеет вид цилиндра высотой 0.1 м, радиусом 0.05 м. Открывающийся угол сканирования по вертикали 60°, по горизонтали 360°, слепая зона 0.2 м, предельная дальность 4.5 м, оптических пучков по вертикали и горизонтали - 25x25 шт. Лидар размещен по центру корпуса модели квадрокоптера. При работе он создает массив координат зарегистрированных точек, узлы с этими координатами помечаются на навигационной сетке как препятствия.

Следовательно, задача сводится к следующему виду:

- необходимо перемещать таргет квадрокоптера по узлам навигационной сетки к цели;
- из текущего узла таргет можно переместить на узел в 6 направлениях: вверх, вниз, влево, вправо, вперед, назад. Так же работает и вариант с 26 направлениями (добавляются все перемещения по диагоналям), однако это существенно снижает производительность симуляции;
- следующий узел считается проходимым для таргета, если в области 0.4x0.4x0.2 м вокруг него нет узлов, содержащих препятствия (выбранная область примерно соответствует габаритам модели коптера с лидаром);
- таким образом, необходимо при помощи алгоритма A* с эвристикой Манхэттена построить маршрут из точки А в точку Б по навигационной сетке и провести по нему таргет коптера.

На сцене симуляции путем размещения стенок был сооружен тестовый полигон с препятствиями, которые не были заранее нанесены на навигационную сетку. Был предложен следующий триггер на перестроение маршрута из текущей точки: если в области следующей точки маршрута, или в области точки маршрута, отстоящей от текущей на 15 позиций в массиве маршрута, обнаружено препятствие – перестроить маршрут. 15 позиций примерно соответствуют половине максимальной дальности зоны действия лидара (1.5 м).

Результат

Квадрокоптер успешно долетел до точки назначения на тестовом полигоне, пролёт записан на видео.

По производительности симуляции можно сделать следующие замечания:

- на видео это не регистрируется, но в зависимости от ситуации скрипту может потребоваться от 40 секунд до 20 минут, чтобы проложить оптимальный маршрут в обход препятствий (в это время симуляция замораживается);
- связана эта задержка с тем, что сама среда симуляции и её Regular API использует не полный ресурс процессора и оперативной памяти (на компьютере автора: 25% ОЗУ и всего 40% CPU, GPU почти не используется) – на данный момент не найдено способа снять эти ограничения;
- у симулятора так же есть возможность взаимодействия среды с Remote API (сторонний код на произвольном языке работает как сервер, и обменивается информацией со средой симуляции, управляя ей) – необходимо повторить эксперимент с использованием данной технологии (на код не будут действовать ограничения ресурсов);
- необходимо переработать триггер перестроения маршрута – существующая в данный момент реализация не проверяет маршрут целиком каждый шаг симуляции, чем не нагружает систему, но также и не обеспечивает заведомой правильности в выборе ситуаций, в которых необходимо перестраивать маршрут;

Выводы.

Алгоритм показал свою пригодность в условиях поставленной задачи – успешно строит маршрут в пространстве в обход препятствий, возвращает информацию о маршруте в пригодном для дальнейшего использования в других вычислениях виде. Требуется переработка триггера перестроения маршрута. В готовом виде, в сочетании с корректным триггером алгоритм может справиться с поставленной задачей, и БПЛА предположительно сможет самостоятельно проложить маршрут на местности в случае недоступности связи с другими агентами роя или коллектива, хотя это займет куда большее время, нежели при наличии готовой карты местности с постоянно поступающей от других агентов информацией о препятствиях. В будущем необходимо провести тест модели с применением технологии Remote API V-REP на языке Python, а в последствии – на реальной физической модели квадрокоптера, оборудованного лидаром.