

УДК 62-503.57

В.Н. Герасимов

СИСТЕМА УПРАВЛЕНИЯ ДВИЖЕНИЕМ МОБИЛЬНОГО РОБОТА В СРЕДЕ С ДИНАМИЧЕСКИМИ ПРЕПЯТСТВИЯМИ

V.N. Gerasimov

THE MOTION CONTROL SYSTEM OF THE MOBILE ROBOT IN ENVIRONMENT WITH DYNAMIC OBSTACLES

Исследованы два метода управления движением мобильного робота в составе навигационной системы: метод эффективного пути и метод динамического окна. Проведено моделирование и получены зависимости результатов работы методов от различных параметров. Предложен алгоритм определения параметров движения динамических препятствий по данным лазерного дальномера.

МОБИЛЬНЫЙ РОБОТ. ПОДВИЖНЫЕ ПРЕПЯТСТВИЯ. УПРАВЛЕНИЕ РОБОТОМ. ЛАЗЕРНЫЙ ДАЛЬНОМЕР.

In the paper two methods of mobile robot motion control are investigated in composition of navigation system: effective path method and dynamic window method. A simulation was performed and dependencies of methods results from various parameters are obtained. The algorithm to determine the parameters of motion from laser rangefinder is offered.

MOBILE ROBOT. MOVING OBSTACLES. ROBOT CONTROL. LASER RANGEFINDER.

Задача автоматического управления мобильным роботом в среде, содержащей подвижные препятствия, в настоящее время является актуальной. В статье рассматриваются и анализируются методы ее решения. Задача ставится следующим образом. Мобильный робот, оснащенный сканирующим лазерным дальномером, находится в помещении. Известна карта данного помещения, построенная при помощи алгоритма SLAM. Приводной уровень системы управления робота выполняет задание вектора скорости $(v, \omega)^T$. Необходимо, управляя вектором скорости, провести робот до заданной целевой точки, избегая столкновения с препятствиями (неподвижными и подвижными).

Существует много способов решения данной задачи, поэтому целью исследования является выбор наиболее эффективных методов: универсальных, надежных, простых в реализации; затем – проведение моделирования и, наконец, их сравнение по результатам моделирования.

Решение поставленной задачи обычно разделяют на два этапа: поиск оптимального маршрута по карте и следование вдоль полученного маршрута. Существует множество алгоритмов поиска оптимального маршрута, в нашем исследовании будет применяться наиболее широко распространенный – A^* (A-звезда). Этот алгоритм достаточно хорошо изучен, поэтому в данной статье не будем его подробно рассматривать.

Задача следования вдоль маршрута имеет несколько подходов к решению, взятых из теории автоматического управления. Существуют различные системы управления, построенные на разных принципах:

- 1) СУ с линейными регуляторами;
- 2) оптимальное управление;
- 3) нечетко-логический контроллер;
- 4) нейросетевой контроллер.

Рассмотрим подробнее два первых варианта. В первом случае задача сводится к расчету опорной точки – положению на плоскости, к которому должен стремиться



ся робот, а также реализации внешнего контура управления приводного уровня робота, обрабатывающего заданное положение на плоскости. Одним из наиболее успешных методов вычисления опорной точки является метод эффективного пути [1]. Суть метода заключается в следующем. В качестве номинальной опорной точки принимается второй излом кусочно-линейной траектории. Это предположение основано на факте, что обнаружение первого изменения направления траектории, которое дало бы опорную точку с дискретным шагом в 45° относительно положения робота, недостаточно для того, чтобы определить тенденцию дальнейшего изменения траектории. Кроме того, вводятся ограничения на длину вектора эффективного пути: сверху с учетом максимально возможного времени беспрепятственного движения и снизу, учитывая минимальный тормозной путь. Таким образом, длина эффективного пути адаптивно увеличивается при повышении скорости робота, и наоборот.

Перейдем к системе управления с линейными регуляторами. Ее особенность заключается в том, что она содержит два параллельных контура слежения: курс на опорную точку и расстояние до нее, причем управление в контуре расстояния зависит от сигнала рассогласования в контуре курса. Такой подход объясняется тем, что движение к цели имеет смысл только тогда, когда курс робота близок к курсу на опорную точку. При этом, чем меньше ошибка, тем выше скорость перемещения.

Предложенный выше вариант управления мобильным роботом позволяет двигаться в среде с подвижными препятствиями, рассматривая их как статические в каждый отдельный момент времени. Однако оригинальный метод эффективного пути не предполагает учет подвижных препятствий и, следовательно, не гарантирует эффективное маневрирование в динамической среде без столкновений. Поэтому он был модифицирован так, чтобы изменять положение опорной точки в зависимости от наличия препятствий и параметров их движения [2]. Очевидно, что для определения опорной точки в динамической среде

необходимо экстраполировать траекторию движения как препятствий, так и робота. В обозначенной статье данный алгоритм описан подробно, здесь же укажем только ключевой момент: если при экстраполяции траектории пересекаются, то необходимо еще раз выполнить планирование маршрута на карте с новыми препятствиями, расположенными в точках пересечения траекторий, а также сдвинуть опорную точку в зону, свободную от столкновений.

Теперь, имея представление о реализации метода эффективного пути, можно провести моделирование. В ходе моделирования проверялись следующие характеристики методов эффективного пути (оригинального и модифицированного для учета динамических препятствий):

L_{path} – длина пройденного роботом пути;

T_{path} – время от начала движения до достижения роботом целевой точки;

D_{max} – максимальное отклонение траектории робота от спланированного маршрута;

траектория робота (для визуальной оценки).

При этом были получены зависимости этих характеристик от параметров методов (максимально допустимые скорости и ускорения, коэффициенты регуляторов в контуре управления и т. п.) и количества препятствий.

Для сравнения метода с учетом подвижных препятствий и без него проведено моделирование с различным числом препятствий от нуля до десяти. На рис. 1 а, б представлены зависимости интересующих нас характеристик от числа препятствий и траектории движения робота для каждого случая без учета подвижных препятствий. Как можно заметить, при увеличении числа препятствий растет время достижения цели, длина пройденного пути. Результаты работы алгоритма, использующего метод эффективного пути с учетом подвижных препятствий, рассмотрим на рис. 1 в, г. Визуально оценивая полученные траектории, уже можно сделать вывод об эффективности данного метода в отличие от предыдущего, однако, примем к сведению объек-

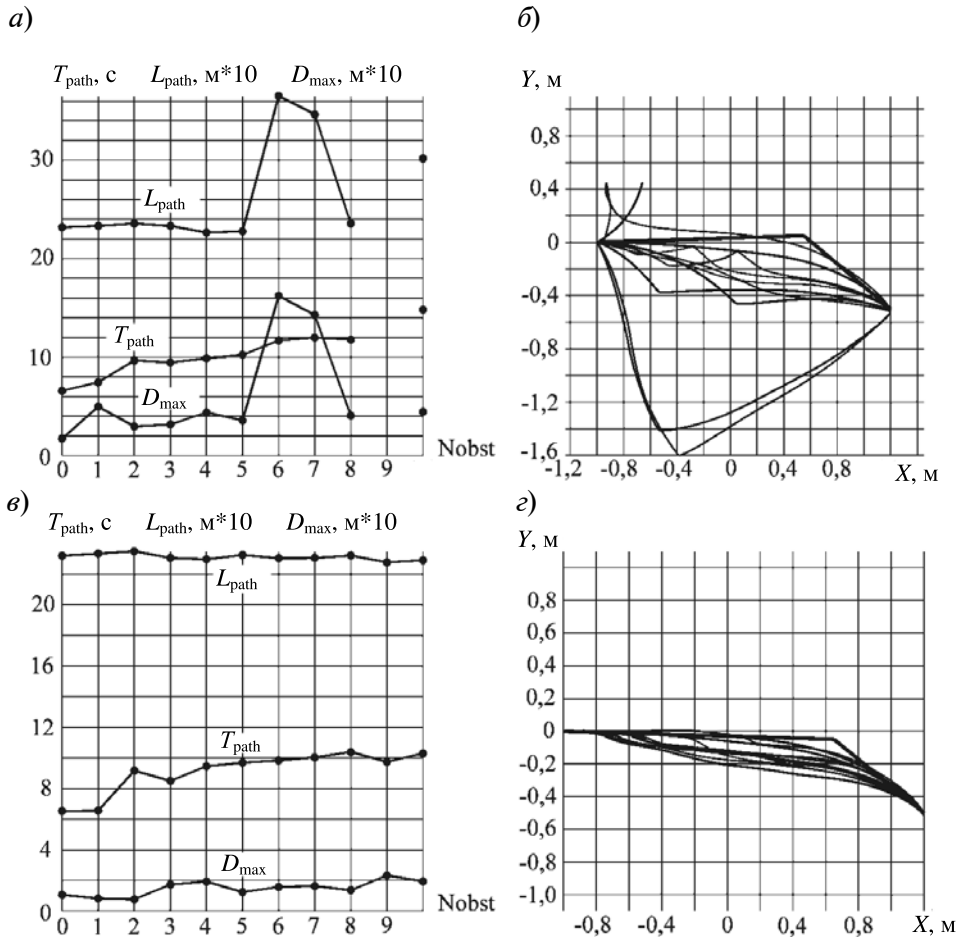


Рис. 1. Результаты моделирования метода эффективного пути

тивную оценку и обратимся к графикам. В данном случае длина пройденного пути практически постоянна, время достижения цели увеличивается незначительно с ростом числа препятствий, а также отклонение от траектории намного меньше, чем в предыдущем случае.

Итак, проведя моделирование, можно сделать вывод о работоспособности метода. Также подтверждается, что модификация для учета подвижных препятствий повысила его надежность и эффективность.

Перейдем к методам оптимального управления. Одна из наиболее успешных реализаций — это универсальный метод динамического окна [3], суть которого заключается в выборе оптимального вектора скорости из допустимой области пространства скоростей, которая получила название «динамическое окно»:

$$V_r = V_s \cap V_a \cap V_d,$$

где V_s — допустимые скорости, ограничены $[0, v_{max}]$, $[-\omega_{max}, \omega_{max}]$; V_a — безопасные скорости (движение с ними не приводит к столкновению); V_d — динамические ограничения (по ускорениям).

Этот метод позволяет непосредственно учесть динамику мобильного робота. В основе решения лежит поиск экстремума функционала в динамическом окне:

$$(v^* \omega^*)^T = \arg \text{extr}(G(v, \omega)),$$

где $G(v, \omega)$ — критерий оптимальности — функционал, учитывающий степень совпадения траектории робота со спланированным маршрутом, а также расположение и траектории движения препятствий.

В оригинальной статье предлагается максимизировать функционал следующего вида:

$$G(v, \omega) = \sigma(\alpha \text{ heading}(v, \omega) + \beta \text{ dist}(v, \omega) + \gamma \text{ velocity}(v, \omega)),$$

где $\text{heading}(v, \omega)$ – функционал курса на целевую точку; $\text{dist}(v, \omega)$ – функционал расстояния до ближайшего препятствия; $\text{velocity}(v, \omega)$ – функционал скорости движения робота; α, β, γ – весовые коэффициенты; σ – нормирующий коэффициент.

В работе [4] метод динамического окна скомбинирован с методом эффективного пути и предложен комплексный критерий, учитывающий близость реальной траектории робота к вектору эффективного пути, а также расстояние до препятствия:

$$G(v, \omega) = \lambda \vartheta_{\text{clear}} + (1 - \lambda) \vartheta_{\text{path}},$$

где ϑ_{clear} – функционал длины пути, свободного от препятствий; ϑ_{path} – функционал близости траектории; λ – весовой коэффициент.

Для учета подвижных препятствий, траектории, которые приводят к столкновению, убираются из пространства допустимых скоростей, однако необходимо перепланировать маршрут с препятствиями в новых местах.

Для проверки работоспособности проведено моделирование (рис. 2). Из рисунка видно, что не все эксперименты завершились успешно (рис. 2 а, б). В случае с допущением статичности, в основном, из-за достаточно большого отклонения от спланированного маршрута. Во втором случае (рис. 2 в, г) ошибки реже, но имеются. Как показали исследования, столкновение происходит, когда пространство допустимых скоростей становится пустым, т. е. нет траектории, двигаясь вдоль которой можно избежать столкновения.

Итак, метод динамического окна также является работоспособным, однако менее

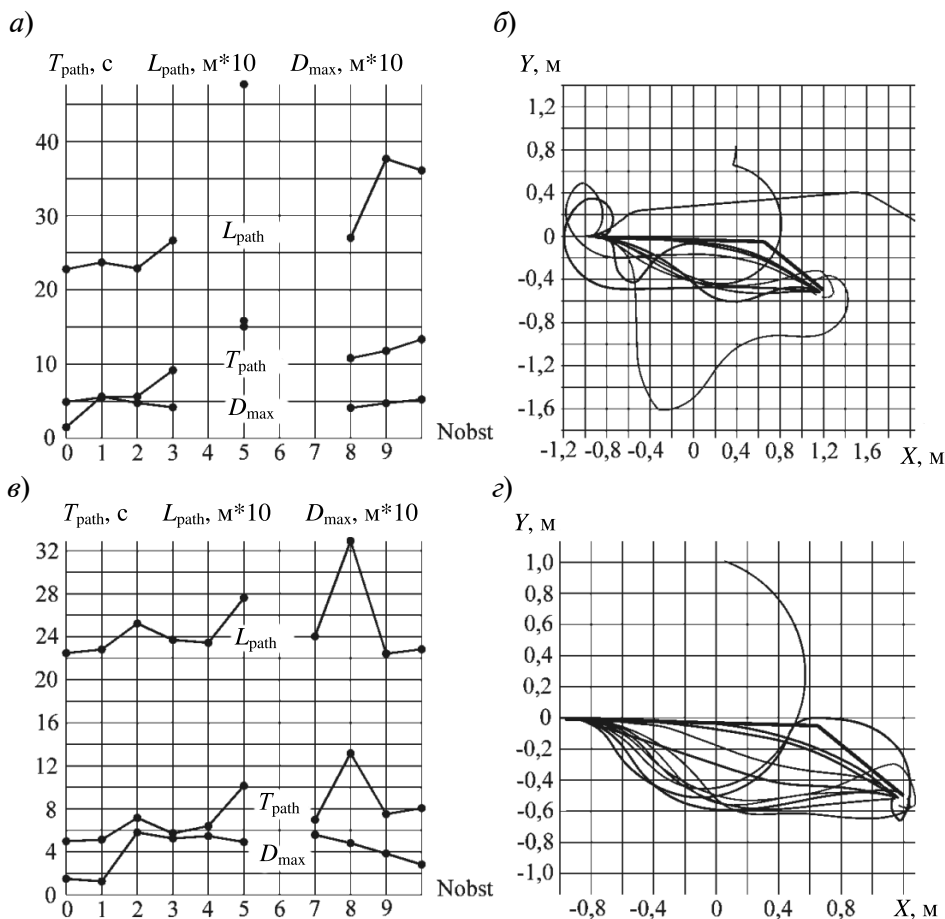


Рис. 2. Результаты моделирования метода динамического окна

надежным, с необходимостью доработки. Преимущество данного метода в его универсальности: можно задать критерий оптимальности движения, необходимый для конкретной задачи. Это может быть минимальное время, кратчайший путь, минимальное отклонение от спланированного маршрута, минимальные энергозатраты и т. п.

Сравним оба метода по результатам моделирования. Метод динамического окна показал себя менее надежным, зато робот достигает цели за более короткое время. Еще один недостаток метода динамического окна — вычислительная сложность.

Перейдем к задаче определения параметров движения препятствий. Эта задача сложная, имеющая множество способов решения в зависимости от типа сенсора, типа входных данных, априорной информации об объекте и т. п. В нашем случае в качестве сенсора используется лазерный дальномер с угловым разрешением 1° и разрешением по дальности 10 мм. Входными данными является массив точек, описывающий рельеф помещения в плоскости измерения. В качестве априорной информации имеется карта помещения, построенная с помощью алгоритма NDT (Normal Distributions Transform — преобразование нормальных распределений) [5]. Также алгоритм SLAM дает текущее положение мобильного робота в системе координат, связанной с помещением, и оценку скорости движения робота.

В результате решения данной задачи необходимо получить список подвижных объектов со следующими параметрами:

- текущее положение;
- текущий вектор скорости;
- геометрические размеры объекта.

Рассмотрим ограничения, которые позволяют значительно упростить решение данной задачи, но при этом решение должно успешно применяться в ранее изложенном алгоритме обхода подвижных препятствий:

1) все препятствия типа «мобильный робот», т. е. подчиняются уравнениям движения трехколесного мобильного робота и достаточно инерционные, т. е. не могут скачкообразно изменить скорость;

2) размеры объекта определяются диаметром габаритной окружности, построенной

таким образом, чтобы охватывать все точки на скане, описывающие данный объект;

3) курс движения препятствия (угол поворота относительно абсолютной системы координат) определяется направлением вектора его линейной скорости;

4) если скорость перемещения препятствия невелика по сравнению с максимальной скоростью мобильного робота, будем считать его статическим в каждый момент времени, т. к. алгоритм обхода статических препятствий допускает медленные изменения расположения препятствий.

Итак, учитывая эти ограничения, необходимо, имея карту помещения, текущий скан и собственное положение робота, найти следующие параметры каждого подвижного препятствия:

- диаметр габаритной окружности W ;
- координаты центра этой окружности (x, y) ;
- курс φ ;
- вектор скорости $(v, \omega)^T$.

Для решения поставленной задачи предлагается использовать следующий алгоритм:

1. Проинициализировать список подвижных объектов.
2. Ждать новый скан.
3. Из всех точек скана найти такие, которые не попадают на объекты на NDT-карте.
4. Провести кластеризацию этих точек, составить список кластеров.
5. Выявить изолированные точки и удалить их.
6. Для каждого объекта из списка кластеров:
7. Определить диаметр габаритной окружности и ее центр.
8. Из списка объектов найти наиболее близкий по положению и размеру.
9. Если окружности объекта и кластера не пересекаются, создать новый объект и добавить его в список.
10. В противном случае обновить оценку скорости и курса по новому положению объекта.
11. Продолжить цикл по кластерам, перейти к п. 5.
12. Перейти к п. 2.

Рассмотрим подробнее основные пункты предложенного алгоритма. Определение точек скана, принадлежащих подвижным препятствиям, проводится с использованием NDT-карты, которая представляет собой функцию, являющуюся суммой функций нормального распределения точек скана в ячейках карты. Распределение в одной ячейке представляется следующим выражением:

$$p(\bar{x}) = \frac{1}{2\pi\sqrt{|\Sigma|}} \exp\left(-\frac{(\bar{x} - \bar{\mu})^T \Sigma^{-1} (\bar{x} - \bar{\mu})}{2}\right). \quad (1)$$

На такой карте в местах, где находятся статические препятствия, функция достигает локального максимума, при удалении от препятствий функция монотонно убывает до нуля, а в ячейках, где отсутствуют препятствия, функция тождественно равна нулю.

Для проверки точек скана необходимо определить значение выражения (1). Очевидно, если результат равен нулю, то данная точка не описывает статическое препятствие, следовательно, если карта актуальна, она описывает подвижное препятствие, т. к. другой альтернативы нет. Однако такая простая оценка не всегда эффективна. Существует корреляция между значением функции и принадлежностью точки к подвижному препятствию, но прямой зависимости нет. Рассмотрим различные ситуации.

1. Функция обращается в нуль в данной точке, следовательно:

1) точка относится к подвижному препятствию, если карта актуальна;

2) точка относится к неподвижному препятствию, если есть неточность в карте.

2. Функция близка к нулю, следовательно:

1) точка относится к подвижному препятствию, если оно находится в непосредственной близости от статического;

2) точка относится к неподвижному препятствию из-за погрешности измерения в текущий момент или в момент построения карты.

3. Функция близка к единице или равна единице, следовательно:

1) точка относится к подвижному препятствию, если препятствие не перемещалось во время построения карты, а затем начало движение;

2) точка относится к неподвижному препятствию, в противном случае.

Как видно, по значению функции нельзя однозначно определить принадлежность точки подвижному препятствию, однако, необходимо принять во внимание допущения, сформулированные ранее, а именно допущение 4. При небольшой скорости перемещения объекта и одновременном построении карты мы получим случай 3, и учитывая данное допущение, отнесем точку к неподвижному препятствию. В первых двух случаях привязка точки к статическому препятствию менее вероятна, чем к подвижному, и связано это с неточностями и погрешностями измерений. Кроме того, случайное определение статического препятствия в качестве динамического добавит вычислительную нагрузку, но в дальнейшем алгоритм не будет его обрабатывать, оценив скорость его перемещения, которая в идеале равна нулю. Также если точка, неверно отнесенная к подвижному препятствию, будет в своей окрестности единственной, необходимо исключить ее из рассмотрения.

Подведя итог рассуждениям, можно сделать следующий вывод: *точка скана считается принадлежащей подвижному препятствию, если значение функции нормального распределения близко к нулю в данной точке.* На практике необходимо задать некоторый порог, который эмпирически определяется, исходя из вероятности допустимого ошибочного определения.

Следующий важный этап обработки полученных точек – это кластеризация. Основная цель кластеризации – сгруппировать точки по некоторому признаку. Также данный алгоритм позволяет выявить изолированные точки. Нам необходимо кластеризовать точки, принадлежащие одному объекту. Имея только дальнометрический скан, очевидным способом является классификация по евклидовому расстоянию между соседними точками. Исходя из характеристик лазерного дальномера, приближенно рассчитаем порог допустимо-

го расстояния между соседними точками. Найдем расстояние между соседними точками скана на объекте, находящемся в 3 м от робота, учитывая угловое разрешение лазерного дальномера в 1° ,

$$d_{\max} = L \cdot \Delta\varphi \approx 3 \cdot 0,0175 \approx 0,05 \text{ м.}$$

Данный порог следует считать исходным значением для эмпирического подбора, т. к. характер принадлежности точек одному объекту вероятностный.

После получения кластеров точек на следующем этапе необходимо определить параметры габаритной окружности. На расчетной схеме (рис. 3) показаны точки скана, относящиеся к некоторому подвижному препятствию $\{P_i : i = 1, \dots, n\}$, габаритная окружность с центром в точке C и радиусом R , которая описывает эти точки и которую требуется найти. В точке O находится начало координат лазерного дальномера; ρ – расстояние до крайней точки скана; L – расстояние до центра C габаритной окружности; a – половина расстояния между крайними точками скана P_1 и P_n .

Расчет основан на том факте, что видимая дуга окружности на концах касается линии луча лазерного дальномера. Обозначим угол обзора объекта из точки O как 2β . Тогда половина этого угла равна углу между радиусом CP_1 и хордой, соединяющей крайние точки P_1P_n . Выражая неизвестные параметры через измеренные, получим выражения для расчета расстояния до центра габаритной окружности объекта, а также ее радиуса:

$$L = \frac{\rho^2}{\sqrt{\rho^2 - a^2}}, \quad R = \sqrt{L^2 - \rho^2}.$$

Положение точки C найдем, отложив на биссектрисе угла P_1OP_n расстояние L .

Итак, получив положение подвижного препятствия и его размер, необходимо проверить, имеется ли похожий объект в списке. Проверка проводится по двум критериям:

- расстояние между центрами объекта в списке и проверяемого не должно превышать некоторого порога, который определяется максимальной скоростью движения препятствия и частотой получения сканов от лазерного дальномера;
- габаритный размер объекта в списке отличается от проверяемого не более, чем на заранее определенное значение.

Если этим критериям удовлетворяют несколько объектов в списке, выбирается наиболее близкий по расстоянию и размеру. Кроме того, при этом можно использовать предсказание положения объекта из списка, если имеется оценка его скорости и курса.

Для использования в данном алгоритме предлагается упрощенный вариант оценки скорости, т. к. много вычислительных ресурсов используется для решения рассмотренных выше проблем. Сначала определяем разность положений нового объекта и старого, затем, зная период измерений h , находим грубую оценку скоростей по двум координатам:

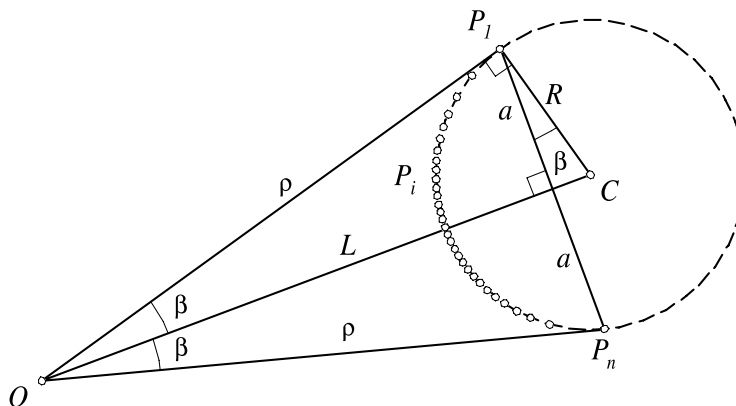


Рис. 3. Расчетная схема определения параметров габаритной окружности

$$\dot{x}_i = \frac{x_i - x_{i-1}}{h}, \dot{y}_i = \frac{y_i - y_{i-1}}{h}.$$

Учитывая инерционность препятствия, что было допущено при постановке задачи, к грубой оценке скорости применим ФНЧ (фильтр нижних частот) для снижения шума дискретного аналога дифференцирования, используя следующие рекуррентные выражения:

$$\hat{x}_i = \lambda \hat{x}_{i-1} + (1 - \lambda) \dot{x}_i, \hat{y}_i = \lambda \hat{y}_{i-1} + (1 - \lambda) \dot{y}_i,$$

где \dot{x}_i, \dot{y}_i – текущие измеренные скорости перемещения по каждой координате; \hat{x}_i, \hat{y}_i – фильтрованные оценки скорости; $\lambda \in (0; 1)$ – характеристика фильтра, определяющая частоту среза.

Теперь, подставляя в стандартные уравнения кинематики трехколесного мобильного робота полученные оценки вместо \dot{x}, \dot{y} , получим следующие выражения для расчета оценки текущего вектора скорости $(\hat{v}_i, \hat{\omega}_i)^T$:

$$\hat{v}_i = \sqrt{\hat{x}_i^2 + \hat{y}_i^2}, \varphi_i = \arctg\left(\frac{\hat{y}_i}{\hat{x}_i}\right),$$

$$\hat{\omega}_i = \frac{\varphi_i - \varphi_{i-1}}{h}.$$

Из-за повторного дифференцирования при расчете $\hat{\omega}$ необходимо также пропустить это значение через ФНЧ:

$$\hat{\omega}_i = \lambda_\omega \hat{\omega}_{i-1} + (1 - \lambda_\omega) \omega_i.$$

СПИСОК ЛИТЕРАТУРЫ

1. Seder, M. Motion control of mobile robots in indoor dynamic environments [Text] / M. Seder, I. Petrovic, K. Macek // Proc. of the Internat. Conf. on Electrical Drives and Power Electronics. – Dubrovnik, Croatia, 2005.
2. Герасимов, В.Н. Решение задачи управления движением мобильного робота при наличии динамических препятствий [Текст] / В.Н. Герасимов, Б.Б. Михайлов // Вестник МГТУ им. Н.Э. Баумана. Приборостроение. – 2012. – С. 83–92.
3. Fox, D. The dynamic window approach to

Итак, получены все необходимые параметры подвижных объектов для метода управления мобильным роботом в среде с подвижными препятствиями. Однако у данного алгоритма есть существенные недостатки, связанные с ограничениями по типу подвижных препятствий, их форме, модели движения. Несмотря на эти недостатки, предложенный алгоритм позволяет поставить эксперимент на реальном роботе в реальном помещении для оценки работоспособности метода управления движением мобильного робота в динамической среде.

Таким образом, при исследовании методов управления движением были получены следующие результаты.

Метод эффективного пути работоспособен и надежен, прост в реализации, не требователен к вычислительным ресурсам, но не является универсальным.

Метод динамического окна также работоспособен, менее надежен (в текущей реализации), требует больше вычислительных ресурсов, но робот достигает цели за более короткий промежуток времени в среде с большим количеством подвижных препятствий; также метод является универсальным в плане выбора критерия оптимальности.

Предложен алгоритм определения параметров подвижных объектов по дальнометрическому изображению и карте помещения.

collision avoidance [Text] / D. Fox, W. Burgard, S. Thrun // Robotics & Automation Magazine IEEE. – 1997. – Т. 4. – P. 23–33.

4. Seder, M. Dynamic window based approach to mobile robot motion control in the presence of moving obstacles [Text] / M. Seder, I. Petrovic // Proc. of the 2007 IEEE Internat. Conf. on Robotics & Automation. – 2007. – P. 1986–1991.

5. Biber, P. The normal distributions transform: a new approach to laser scan matching [Text] / P. Biber, W. Strasser // Proc. of the 2003 IEEE/RSJ Internat. Conf. – 2003. – Т. 3. – P. 2743–2748.

REFERENCES

1. Seder M., Petrovic I., Macek K. Motion control of mobile robots in indoor dynamic environments / Proc. of the Internat. Conf. on Electrical Drives and Power Electronics. – Dubrovnik, Croatia, 2005.

2. Gerasimov V.N., Mikhailov B.B. Reshenie zadachi upravleniya dvizheniem mobilnogo robota pri nalichii dinamicheskikh prepyatstviy [The solution of mobile robot motion control problem in presence

of dynamic obstacles] / Vestnik MGTU im. N.E. Baumana. Priborostroenie. – 2012. – S. 83–92. (rus)

3. **Fox D., Burgard W., Thrun S.** The dynamic window approach to collision avoidance / Robotics & Automation Magazine IEEE. – 1997. – Т. 4. – P. 23–33.

4. **Seder M., Petrovic I.** Dynamic window based

approach to mobile robot motion control in the presence of moving obstacles / Proc. of the 2007 IEEE Internat. Conf. on Robotics & Automation. – 2007. – P. 1986–1991.

5. **Biber P., Strasser W.** The normal distributions transform: a new approach to laser scan matching / Proc. of the 2003 IEEE/RSJ Internat. Conf. – 2003. – Т. 3. – P. 2743–2748.

ГЕРАСИМОВ Владимир Николаевич – аспирант Московского государственного технического университета имени Н.Э. Баумана.

105005, Россия, Москва, ул. 2-я Бауманская, д. 5.

E-mail: vovka_ok@list.ru

GERASIMOV Vladimir N. *Bauman Moscow State Technical University.*

105005, 2-ya Baumanskaya Str. 5, Moscow, Russia.

E-mail: vovka_ok@list.ru