

УДК 004.896+535.8

Алгоритм описания траектории мобильного робота по данным визуального одометра для автоматического возвращения к оператору

Девятериков Е. А.^{1,*}

[*retegor@mail.ru](mailto:retegor@mail.ru)

¹МГТУ им. Н.Э. Баумана, Москва, Россия

В работе представлен алгоритм автономного описания, аппроксимации и сохранения траектории мобильного робота по данным визуального одометра для автоматического возвращения к оператору. Данные аппроксимируются двумерной кусочно-полиномиальной функцией (сплайном) 3-й степени. Визуальная одометрия – метод определения координат с помощью технического зрения, основанный на измерении смещения ключевых точек пространства, информация о которых получается из анализа последовательности изображений. Такая система работает в заранее не подготовленной, недетерминированной среде, нечувствительна к проскальзыванию колес. В работе приводятся основные соотношения для описания траектории робота, результаты проверки алгоритма и моделирования возврата робота.

Ключевые слова: мобильный робот, описание, сохранение траектории, пути, визуальный одометр, автоматическое возвращение, возврат

Введение

Для дистанционно управляемого мобильного робота в условиях недостаточной видимости, при отсутствии карт территории, задержек или полной потери связи требуется применять систему автоматического возвращения к оператору. Такие системы основаны на подсистеме автономной навигации, определяющей собственные координаты мобильного робота. Проблема возникает в условиях недетерминированной среды, когда нет точек отсчета, например стационарных маяков, и экстремальных условиях, в которых средства автономной навигации не всегда позволяют измерять текущие координаты робота достаточно точно. В настоящее время для навигации широко используются [1]: энкодеры, инерциальные системы, GPS и дальномеры. Но все перечисленные устройства имеют свои недостатки. Использование энкодеров невозможно при проскальзывании колес. Инерциальные системы накапливают ошибку. GPS недостаточно точна для локальной навигации и неустойчива в закрытых помещениях. Система с дальномером имеет высокую стоимость, сложна и требует наличие объектов в зоне видимости.

Альтернативой этим методам является автономная оценка собственного перемещения робота по изображениям с бортовых телекамер – визуальная одометрия [2, 3, 4], которая не имеет перечисленных недостатков. Метод основан на измерении смещения ключевых точек в пространстве, информация о которых получается из анализа последовательности стереоизображений системы технического зрения. Такая система может работать в недетерминированной среде, позволяет решать сопряженные задачи, такие, как построение карты помещения. В ходе предыдущей работы [5, 6, 7] было установлено, что данные визуальной одометрии могут быть использованы для автоматического управления мобильным роботом, в т. ч. для решения задачи возвращения к оператору, но имеют следующие особенности: относительно низкая частота поступления данных 10 Гц высокая дисперсия; зависящая от условий; относительный принцип навигации [8]. Эти особенности не позволяют эффективно использовать методы фильтрации, описания и сохранения траектории, применяемые для других датчиков [1, 8]. Фильтр Калмана, например, при таких характеристиках вносит слишком высокую задержку.

Поэтому целью данной работы является разработка и реализация алгоритма описания траектории мобильного робота по данным визуального одометра для автоматического возвращения к оператору. Для достижения этой цели были поставлены и решены следующие задачи:

- разработать алгоритм автономного описания траектории мобильного робота по данным визуальной одометрии для автоматического возвращения робота к оператору;
- выполнить моделирование алгоритма, проверить возможность его применения.

1. Алгоритм описания траектории и возвращения

С учетом особенностей данных визуального одометра были сформулированы требования к алгоритму: пространственная аппроксимация полученных координат; аддитивное сохранение данных, по мере поступления (а большинство алгоритмов построения сплайна, например, предполагают наличие всех данных); работа в реальном масштабе времени. На основе этих требований был разработан следующий алгоритм описания траектории и возвращения мобильного робота, предполагающий 2 режима работы:

1. Аппроксимация траектории робота в виде кусочно-полиномиальной функции (сплайна) [5].
 - 1.1. Зафиксировать начальные координаты робота $p_0 = (x_0, y_0)$ (рис. 1).
 - 1.2. Получить от визуального одометра текущие координаты $d_i = (x_i, y_i)$ мобильного робота.
 - 1.3. Аппроксимировать полученные точки кривой $f_{x,y}(t)$ 3-го порядка методом наименьших квадратов (МНК).
 - 1.4. Рассчитать максимальное отклонение точек $d_i = (x_i, y_i), i = 0..N$ от кривой $f_{x,y}(t)$.
 - 1.5. Если отклонение меньше заданного E , перейти к 1.2, иначе зафиксировать участок сплайна, построенный по всем точкам, кроме последней $d_i = (x_i, y_i), i = 0..(N - 1)$.
 - 1.6. Выполнить естественную параметризацию кривой $f_{x,y}(t)$.

- 1.7. Найти и удалить петли сплайна, перейти к 1.1.
2. Возвращение робота по сохраненной траектории.
 - 2.1. Найти ближайшую к роботу точку сплайна p_r .
 - 2.2. Найти точку p_f , которая по координате длины сплайна ближе к его началу на величину S_0 .
 - 2.3. Двигаться в т. p_f с линейной скоростью $V = k_v |p_f p_r|$ и угловой скоростью $\omega = k_\omega \alpha$, где α - угол курса робота на точку p_r .
 - 2.4. Если расстояние до начальной точки сплайна больше заданного, перейти к 2.1.

2. Аппроксимация данных визуального одометра кривыми 3-го порядка

Алгоритм был реализован для полиномов 3-й степени в параметрической форме:

$$\begin{cases} x = a_3 t^3 + a_2 t^2 + a_1 t + a_0 = \sum_{m=0}^3 a_m t^m \\ y = b_3 t^3 + b_2 t^2 + b_1 t + b_0 = \sum_{m=0}^3 b_m t^m \end{cases} \quad (1)$$

При таком описании [9] кубическая кривая может быть повернута на любой угол, в отличие от одномерной формы, когда $x = t + x_0$. Не возникает случай плохой обусловленности и бесконечных коэффициентов. Использование полинома третьей степени позволяет ввести одновременно до 3-х ограничений (рис. 3), таких как: начальная точка p_0 , конечная точка p_1 и 1-я производная в точке p_0 .

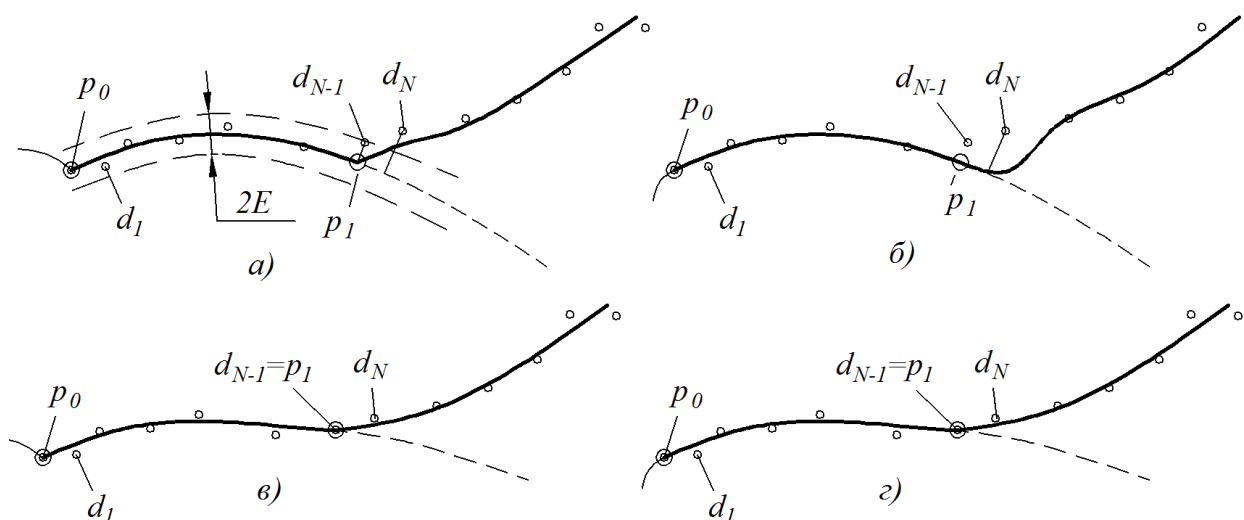


Рис. 1. Участки сплайна, построенные с различными начальными условиями: а) – одна начальная точка, б) – одна начальная точка и производная, в) – две точки, г) – две точки и производная.

Будем независимо искать решение для $x(t)$ и $y(t)$ с помощью МНК, что близко к минимизации расстояний от точек до кривой. Параметр t вводится искусственно, в качестве него выбрана длина ломаной $p_0 d_1 d_2 \dots d_i$. Для каждой поступившей на вход точки d_i рассчитывается $t_i = t_{i-1} + |d_i d_{i-1}|$, $d_i = (x, y, t_i)$. Начальная точка $p_0 = (x_0, y_0, 0)$. В-первых, такой параметр позволяет сохранить порядок следования входных точек. Во-вторых, при небольшом шуме данных $t \sim s$, где s – длина дуги сплайна, что позволит вы-

полнить естественную параметризацию без изменения формы кривой. Рассмотрим варианты аппроксимации с различными начальными условиями.

Одна начальная точка. Введем минимальное ограничение на аппроксимируемую функцию, начальную точку $p_0 = (x_0, y_0)$:

$$x_0 = a_3 t_0^3 + a_2 t_0^2 + a_1 t_0 + a_0 \Rightarrow a_0 = x_0 - a_3 t_0^3 - a_2 t_0^2 - a_1 t_0.$$

Подставив это выражение в (1) получим:

$$(x_i - x_0) = a_3(t_i^3 - t_0^3) + a_2(t_i^2 - t_0^2) + a_1(t_i - t_0), i = 1..N.$$

Численно решая полученную систему уравнений находим коэффициенты a_m , аналогично b_m .

Одна начальная точка и 1-я производная. Для скоростных участков строим гладкий сплайн, для этого добавим начальное условие: производные $(\partial x / \partial t)_{p_0} = x'_0$ и $(\partial y / \partial t)_{p_0} = y'_0$ в точке p_0 .

$$\begin{cases} x_0 = a_3 t_0^3 + a_2 t_0^2 + a_1 t_0 + a_0 \\ x'_0 = 3a_3 t_0^2 + 2a_2 t_0 + a_1 \end{cases} \Rightarrow \begin{cases} a_1 = x'_0 - 3a_3 t_0^2 - 2a_2 t_0 \\ a_0 = x_0 - a_3 t_0^3 - a_2 t_0^2 - x'_0 t_0 - 3a_3 t_0^2 - 2a_2 t_0 \end{cases}$$

Подставляя эти условия в (1), получим переопределенную систему относительно a_3 и a_2 :

$$(x_i - (x'_0 t_i + x_0 - x'_0 t_0)) = (t_i^3 - 3t_i t_0^2 + 2t_0^3)a_3 + (t_i - t_0)^2 a_2, i = 1..N$$

Две точки: начальная и конечная. Требования к быстродействию и устойчивости могут не позволить полностью отфильтровать шум одометра. Тогда аппроксиматор траектории выполняет роль дополнительного фильтра. Если в качестве начальных условий использовать одну начальную точку p_0 (рис. 1 а), то конечная точка d_1 наилучшей в смысле МНК аппроксимации оказывается вне реальной траектории, появляются изломы и другие искажения (рис. 1 б), что влияет на надежность системы. В случае не зашумленных данных целесообразно в качестве дополнительного начального условия использовать последнюю точку данных $d_1 = (x_1, y_1)$. Тогда:

$$\begin{aligned} \begin{cases} x_0 = a_3 t_0^3 + a_2 t_0^2 + a_1 t_0 + a_0 \\ x_1 = a_3 t_1^3 + a_2 t_1^2 + a_1 t_1 + a_0 \end{cases} \Rightarrow \begin{cases} (x_i - x_0) = a_3(t_i^3 - t_0^3) + a_2(t_i^2 - t_0^2) + a_1(t_i - t_0) \\ (x_i - x_1) = a_3(t_i^3 - t_1^3) + a_2(t_i^2 - t_1^2) + a_1(t_i - t_1) \end{cases} \\ (x_i(t_0 - t_1) + t_i(x_1 - x_0) + (x_0 t_1 - x_1 t_0)) = \\ = a_3(t_i^3(t_0 - t_1) + t_i(t_1^3 - t_0^3) + (t_0^3 t_1 - t_1^3 t_0)) \\ + a_2(t_i^2(t_0 - t_1) + t_i(t_1^2 - t_0^2) + (t_1 t_0^2 - t_1^2 t_0)), i = 1..N \end{aligned}$$

Две точки и производная в одной точке. Если к условию прохождения двух точек добавить условие гладкости $(\partial x / \partial t)_{p_0} = x'_0$ и $(\partial y / \partial t)_{p_0} = y'_0$ для скоростных участков траектории, получим

$$\begin{aligned} - \begin{cases} x_0 = a_3 t_0^3 + a_2 t_0^2 + a_1 t_0 + a_0 \\ x_1 = a_3 t_1^3 + a_2 t_1^2 + a_1 t_1 + a_0 \\ x'_0 = 3a_3 t_0^2 + 2a_2 t_0 + a_1 \end{cases} \Rightarrow \begin{cases} (x_0 - x_1) = a_3(t_0^3 - t_1^3) + a_2(t_0^2 - t_1^2) + a_1(t_0 - t_1) \\ a_1 = x'_0 - 3a_3 t_0^2 - 2a_2 t_0 \end{cases} \\ \Rightarrow a_2 = \frac{x_0 - x_1 - x'_0(t_0 - t_1) - a_3(t_0^3 - t_1^3 - 3t_0^2(t_0 - t_1))}{-(t_0 - t_1)^2} \end{aligned}$$

$$(x_i - x_0 - x'_0(t_i - t_0))m_1 - (x_i - x_1 - x'_0(t_i - t_1))m_0 = \\ = a_3 \left[(t_i^3 - t_0^3 - 3t_0^2(t_i - t_0))m_1 - ((t_i^3 - t_1^3 - 3t_0^2(t_i - t_1)))m_0 \right],$$

где $m_1 = t_i^2 - t_1^2 - 2t_0(t_i - t_1)$, $m_0 = t_i^2 - t_0^2 - 2t_0(t_i - t_0)$.

Система имеет решение и хорошо обусловлена, если $(t_0 - t_1)^2 \gg 0$, что гарантируется сеткой входных данных. В этом случае задача МНК решается без обращения матриц в аналитическом виде.

3. Удаление петель

Для решения задачи возврата робота к оператору необходимо очистить сохраненную траекторию от петель [10] – участков сплайна $t_a \leq t < t_b$ с самопересечением в т. $q = (x(t_a), y(t_a)) = (x(t_b), y(t_b))$. Уравнения (1) нельзя представить в виде $y = f(x)$ и аналитически найти пересечение кривых, но это осуществимо численными методами. С учетом требуемой точности разобьем кривые на n отрезков, впишем отрезки в габаритные прямоугольники и найдем пересечение перебором. Способ ресурсоемок, учитывая, что мы должны попарно перебирать все n отрезков всех M сегментов сплайна: $O((n \cdot M)^2)$.

Для оптимизации поиска найдем все экстремумы сегментов сплайна и заменим кривые габаритными прямоугольниками как на рис. 2

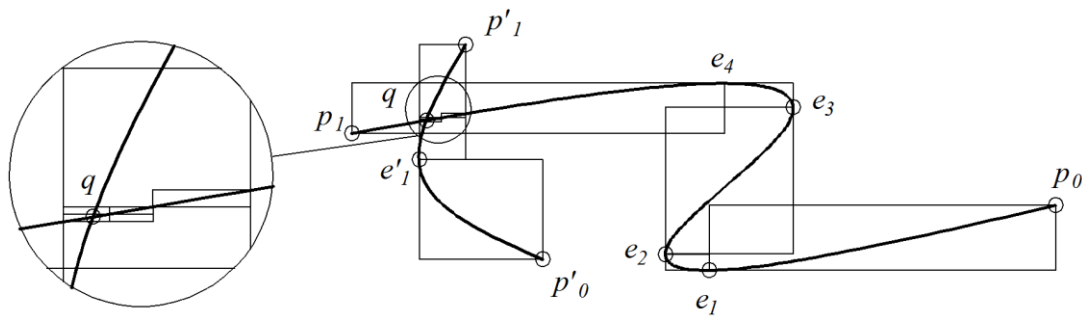


Рис. 2. Двоичный поиск точки пересечения по габаритным прямоугольникам.

Кривая $p_0e_1e_2e_3e_4p_1$ пересекает кривую $p'_0e'_1p'_1$ в точке q . Экстремумы e_1, e_2, e_3, e_4, e'_1 и конечные точки p_0, p'_0 и p_1, p'_1 определяют габариты участков. Затем формируем дерево поиска, узлами которого являются габаритные прямоугольники. Вершина дерева – габариты всего сплайна, каждое поддерево содержит габариты половины сегментов сплайна, затем сплайн разбивается на 4 части и т. д. пока не будут достигнуты прямоугольники экстремумов, например p_1e_4 . Верхняя часть дерева строится один раз и дополняется по

мере поступления новых данных. Теперь организуем двоичный поиск. Найдя прямоугольники, содержащие пересечение, уточняем поиск, продолжая разбивать их на 2 части до заданной точности. В худшем случае, когда сегменты с пересечением не содержат экстремумов, максимальная сложность $O((\log_2(nM))^2)$.

4. Естественная параметризация

Для анализа траектории необходимо определять длину ее участков $s(p_1, p_2)$, ограниченных, точками p_1 и p_2 и находить точки на траектории в виде $p = (f_x(s), f_y(s))$. Формула длины кривой

$$s = \int_a^b \sqrt{(x'(t))^2 + (y'(t))^2} dt$$

с подстановкой (1) в явном виде не интегрируются, что усложняет вычисления. Для оптимизации, численно выполняем естественную параметризацию кривой (1), представляя ее в виде:

$$\begin{cases} x = a'_3 s^3 + a'_2 s^2 + a'_1 s + a'_0 \\ y = b'_3 s^3 + b'_2 s^2 + b'_1 s + b'_0 \end{cases},$$

где s – натуральный параметр (длина дуги кривой).

Для этого новый участок сплайна строится в 3 этапа: 1) Данные аппроксимируются по параметру $t \approx s$. 2) Кривая численно интегрируется и находятся координата длины вдоль этой кривой s для каждой точки данных $p'_i = (x_i, y_i, s_i)$. 3) Строится новая кривая, но уже по точкам $p'_i(x_i, y_i, s_i)$.

5. Проверка алгоритмов

Типовые фрагменты текста, занимающие один абзац, такие как леммы, теоремы, определения и т.п. оформляются специальными стилями:

Предложенные алгоритмы были реализованы в виде программного обеспечения, обрабатывающего показание визуального одометра в реальном времени. Для проверки алгоритмов использовались данные в количестве 505 точек d_i , записанные с одометра в ходе эксперимента, (см. рис. 3а в [6]). Робот в автоматическом режиме двигался из точки $p_0(0, 0)$ в точку F , преодолевая участки длиной 0,5 м и повороты на 90° . На рис. 3 представлены данные одометра d_i и построенные с допуском $E = 50$ мм различными способами аппроксимирующие кривые с узлами в точках p_i .

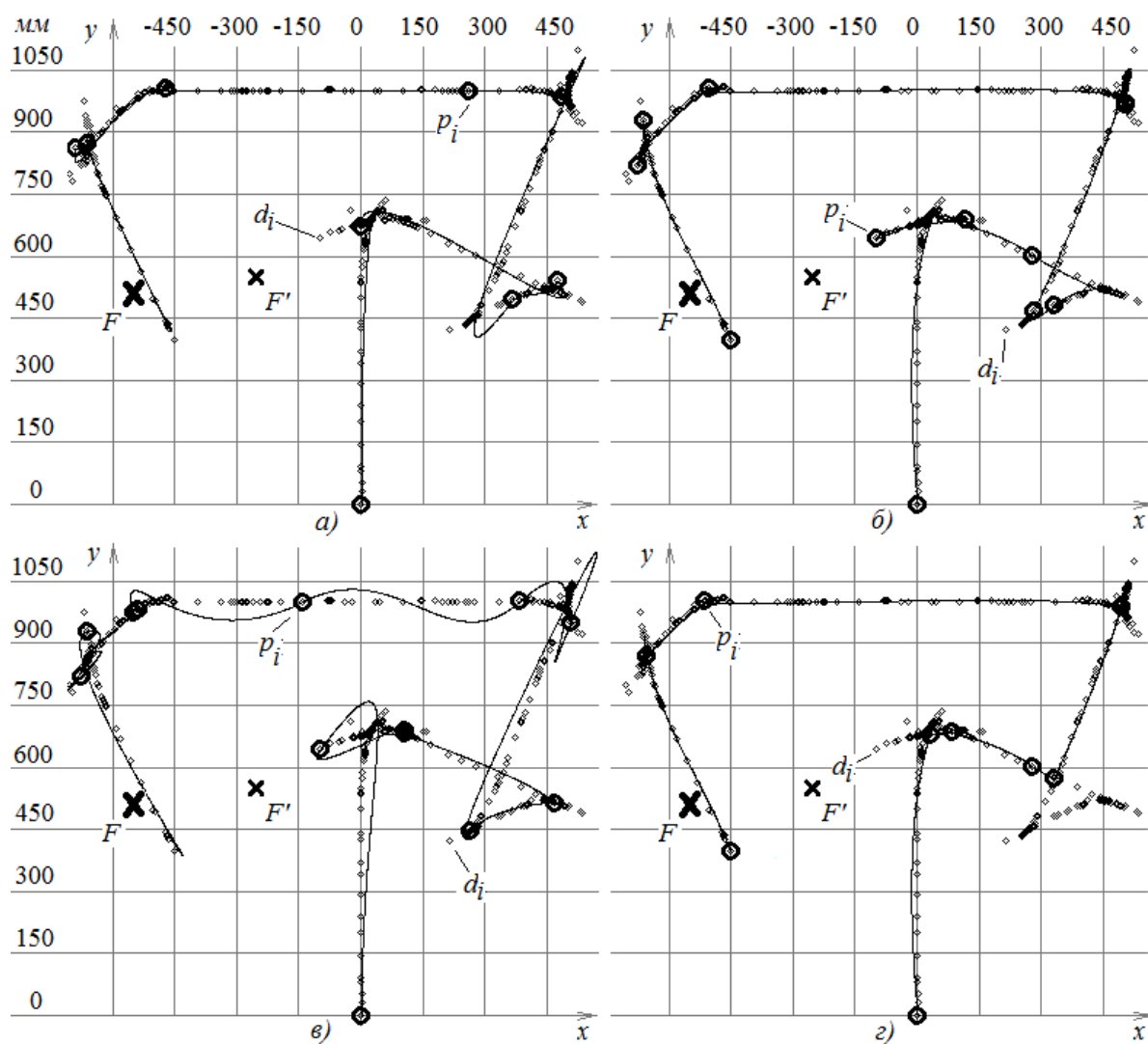


Рис. 3. Результаты описания траектории по реальным данным визуального одометра: а) – одна начальная точка, б) – две точки, в) – две точки и производная, г) – после удаления петель.

На рис. 3г показан сплайн после удаления петель. Предложенный способ построения гладкого сплайна (рис. 3в) порождает множество выпуклостей, проявил ненадежность в ходе отладки и не может быть использован без дополнений. В таблице 1 показано время работы алгоритмов.

Таблица 1. Время вычисления алгоритмов описания пути.

Среднее время полного цикла (с поиском петель с точностью 0,1 мм)	1,0 мс
Максимальное время полного цикла (с поиском петель с точностью 0,1 мм)	3,0 мс
Среднее время поиска петель с точностью 0,1 мм	0,5 мс
Среднее время поиска петель с точностью 0,1 мм в последних 100 циклах	0,5 мс
Максимальное время поиска петель с точностью 0,1 мм	2,5 мс
Среднее время поиска петель с пониженной точностью 1 мм	0,05 мс

Из таблицы видно, что среднее время полного цикла алгоритма описания пути робота составляет 1 мс, что в 100 раз меньше времени цикла визуального одометра [5, 6, 7] при

той же вычислительной мощности. В 505-и циклах построено 9 сегментов сплайна, деградация производительности алгоритма поиска петель не наблюдается.

Для оценки конечной точности возврата робота в исходную точку мы использовали данные эксперимента [6] и модель робота с визуальным одометром [7] в которой учтены характеристики одометра. Полученный сплайн был использован в модели как заданная траектория. На рис. 4 показаны узловые точки p_i сплайна и разброс конечных положений робота v_i и w_i после 40-ка циклов возврата.

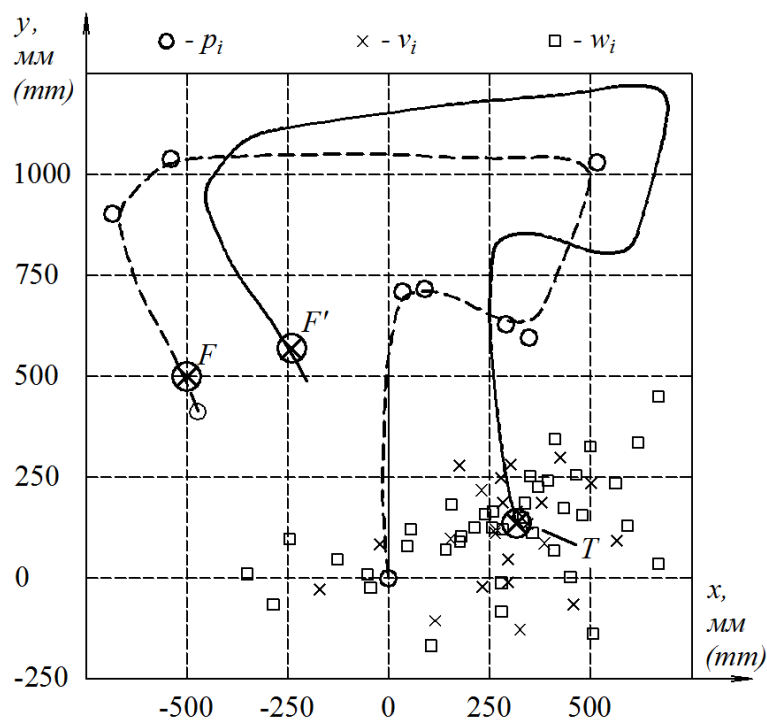


Рис. 4. Разброс конечных положений робота при моделировании возврата.

Пунктиром показана идеальная траектория возврата без учета погрешности одометра и начальной ошибки. Когда одометр зафиксировал точку $F(-500, 500)$, робот реально находился в точке $F' \approx (-250, 550)$ и имел накопленную ошибку угла курса $\alpha = 7^\circ$ (см. [6]). Сплошной линией показана идеальная траектория возврата с учетом начальной ошибки положения. С учетом ошибок визуального одометра получен разброс конечных положений v_i робота. Если для возврата необходимо развернуть робот, увеличивается ошибка курса и разброс конечных положений шире (точки w_i).

Заключение

В ходе работы был спроектирован и программно реализован алгоритмы описания траектории мобильного робота по данным визуального одометра в виде двумерного сплайна 3-го порядка для использования в задаче автоматического возвращения робота к оператору. Выполнена проверка на реальных данных, записанных с визуального одометра, уста-

новленного на мобильном роботе. По полученным кривым выполнено моделирование возврата робота в исходную точку с учетом погрешности одометра. При моделировании отклонение конечного положения робота от исходного не превысило 16% от длины траектории 3,5 м. Проверка быстродействия показала, что алгоритм может быть использован в режиме реального времени. Таким образом, предложенный алгоритм может использоваться с визуальным одометром для автоматического возвращения робота к оператору.

Список литературы

1. Dudek G., Jenkin M. Inertial Sensors, GPS, and Odometry // In: Springer Handbook of Robotics / Ed. by B. Siciliano, O. Khatib. Springer Berlin Heidelberg, 2008. P. 477-580. DOI: [10.1007/978-3-540-30301-5_21](https://doi.org/10.1007/978-3-540-30301-5_21)
2. Nist'er D., Naroditsky O., Bergen J. Visual odometry for ground vehicle applications // Journal of Field Robotics. 2006. Vol. 23, no. 1. P. 3-20. DOI: [10.1002/rob.20103](https://doi.org/10.1002/rob.20103)
3. Scaramuzza D., Fraundorfer F. Visual Odometry [Tutorial] // IEEE Robotics & Automation Magazine. 2011. Vol. 18, is. 4. P. 80-92. DOI: [10.1109/MRA.2011.943233](https://doi.org/10.1109/MRA.2011.943233)
4. Jin Xu, Minyi Shen, Weiqiang Wang, Li Yang, Jilin Liu, Kalyaev I. Robust stereo visual odometry for autonomous rover // Proc. of the 6th WSEAS International Conference on Signal, Speech and Image Processing. 2006.
5. Девятериков Е.А., Михайлов Б.Б. Визуальный одометр // Инженерный журнал: наука и инновации. 2012. № 6. Режим доступа: <http://engjournal.ru/catalog/pribor/robot/249.html> (дата обращения. 01.11.2014).
6. Девятериков Е.А., Михайлов Б.Б. Система управления движением мобильного робота с визуальным одометром // Научно-технические ведомости Санкт-Петербургского государственного политехнического университета. Информатика. Телекоммуникации. Управление. 2013. Вып. 5 (181). С. 103-108.
7. Девятериков Е.А., Михайлов Б.Б. Использование данных визуального одометра в системе управления мобильного робота // Научно-техническая конференция «Экстремальная робототехника 2012»: сб. тр. СПб., 2012. С. 155-163.
8. Holland J.M. Designing Autonomous Mobile Robots: Inside the Mind of an Intelligent Machine. Elsevier Inc., 2004. P. 105-325.
9. Biswas D.K. Spline Guided Path of a Mobile Robot with Obstacle Avoidance Characteristics // Proc. of the 14th National Conference on Machines and Mechanisms (NaCoMM09) (NIT, Durgapur, India, 17-18 December 2009). NaCoMM, 2009. P. 226-271.
10. Sprunk C. Planning Motion Trajectories for Mobile Robots Using Splines. University of Freiburg, Faculty of Applied Sciences, Department of Computer Science, 2008. 110 p.

Algorithm Describing Mobile Robot Path Using the Visual Odometer Data for Automatic Returning to Operator

E.A. Devyaterikov^{1,*}

retegor@mail.ru

¹Bauman Moscow State Technical University, Moscow, Russia

Keywords: mobile robot, trajectory, path description, memorization, visual odometer, automatic returning

Radio-controlled mobile robots operating in degraded visual environment, with uncharted areas and communication loss, need in use of a system of autonomous returning to operator. To solve this task the position and course estimation system is required. Today, the following navigation devices are widely spread: encoders, inertial sensors, GPS, and rangers. However, all above devices are imperfect. Use of encoders is embarrassed with wheels slipping. Inertial systems accumulate an error with time. GPS is non-sufficiently accurate for the purposes of local navigation and unstable in indoor spaces. A ranger-based system is expensive, complicated and requires objects to be in the workspace.

An alternative to these methods is autonomous estimation of robot self-motion using video data from onboard cameras, i.e. visual odometry that is free of above shortcomings. The purpose of this work is to develop and implement an algorithm to describe a mobile robot path using the visual odometer data for automatic returning to operator. For this purpose the following tasks were set and solved:

- to develop the algorithm of memorizing mobile robot path using the visual odometer data for automatic returning to operator;

- to perform algorithm simulation and test it for working capability.

The algorithm of robot path approximation and saving data was developed considering features of visual odometer data. For path approximation the piecewise 3rd order polynomial function was used. It allowed us to impose up to 3 additional restrictions and, being well conditioned and invariant to choosing the coordinate system direction, to hold the order of data sequence. The algorithm also includes the loop removal and natural parameterization.

To test the algorithm were used real data obtained during previous experiments with a real mobile robot. The results of test have shown that the algorithm allows us to describe and save the mobile robot path by visual odometer data in real time. Besides, using the memorized data the mobile robot returning to operator was also simulated. The results confirmed that the proposed

algorithm could be used together with visual odometer to perform automatic mobile robot return to operator.

References

1. Dudek G., Jenkin M. Inertial Sensors, GPS, and Odometry. In: Siciliano B., Khatib O., eds. *Springer Handbook of Robotics*. Springer Berlin Heidelberg, 2008, pp. 477-580. DOI: [10.1007/978-3-540-30301-5_21](https://doi.org/10.1007/978-3-540-30301-5_21)
2. Nist'er D., Naroditsky O., Bergen J. Visual odometry for ground vehicle applications. *Journal of Field Robotics*, 2006, vol. 23, no. 1, pp. 3-20. DOI: [10.1002/rob.20103](https://doi.org/10.1002/rob.20103)
3. Scaramuzza D., Fraundorfer F. Visual Odometry [Tutorial]. *IEEE Robotics & Automation Magazine*, 2011, vol. 18, is. 4, pp. 80-92. DOI: [10.1109/MRA.2011.943233](https://doi.org/10.1109/MRA.2011.943233)
4. Jin Xu, Minyi Shen, Weiqiang Wang, Li Yang, Jilin Liu, Kalyaev I. Robust stereo visual odometry for autonomous rover. *Proc. of the 6th WSEAS International Conference on Signal, Speech and Image Processing*. 2006.
5. Devyaterikov E.A., Mikhaylov B.B. Visual Odometer. *Inzhenernyy zhurnal: nauka i innovatsii = Engineering Journal: Science and Innovation*, 2012, no. 6. Available at: <http://engjournal.ru/catalog/pribor/robot/249.html> , accessed 01.11.2014. (in Russian).
6. Devyaterikov E.A., Mikhaylov B.B. Mobile robot movement control system with visual odometer. *Nauchno-tekhnicheskie vedomosti Sankt-Peterburgskogo gosudarstvennogo politekhnicheskogo universiteta. Informatika. Telekommunikatsii. Upravlenie = St.Petersburg State Polytechnical University Journal. Computer Science. Telecommunication and Control Systems*, 2013, no. 5 (181), pp. 103-108. (in Russian).
7. Devyaterikov E.A., Mikhaylov B.B. Using visual odometer data in the control system of mobile robot. *Nauchno-tekhnicheskaya konferentsiya "Ekstremal'naya robototekhnika 2012": sb. tr.* [Proc. of the Scientific and Technical Conference "Extreme Robotics 2012"]. St. Petersburg, 2012, pp. 155-163. (in Russian).
8. Holland J.M. *Designing Autonomous Mobile Robots: Inside the Mind of an Intelligent Machine*. Elsevier Inc., 2004, pp. 105-325.
9. Biswas D.K. Spline Guided Path of a Mobile Robot with Obstacle Avoidance Characteristics. *Proc. of the 14th National Conference on Machines and Mechanisms (NaCoMM09)*, NIT, Durgapur, India, 17-18 December 2009. NaCoMM, 2009, pp. 226-271.
10. Sprunk C. *Planning Motion Trajectories for Mobile Robots Using Splines*. University of Freiburg, Faculty of Applied Sciences, Department of Computer Science, 2008. 110 p.