



УДК 519.7

П.Н. Брунарский
Москва, Россия

ОБУЧЕНИЕ МОБИЛЬНОГО РОБОТА ДВИЖЕНИЮ ПО ТРАЕКТОРИИ СО СЛОЖНЫМИ ОГРАНИЧЕНИЯМИ

P.N. Brunarskiy
Moscow, Russia

LEARNING MOBILE ROBOT FOR PATH MOTION WITH COMPLEX CONSTRAINTS

Описано применение локального метода адаптации ковариационной матрицы для обучения мобильного робота с частично известными параметрами выполнению различных маневров и движению по траектории со сложными ограничениями. Предложена итеративная модификация алгоритма, ускоряющая процесс обучения для траекторий большой длины.

ОБУЧЕНИЕ. УПРАВЛЕНИЕ ДВИЖЕНИЕМ. ОГРАНИЧЕНИЯ. АДАПТАЦИЯ. МОБИЛЬНЫЙ РОБОТ. ОПТИМИЗАЦИЯ. АДАПТАЦИЯ КОВАРИАЦИОННОЙ МАТРИЦЫ.

This paper describes the application of a local method of the covariance matrix adaptation for learning mobile robot with a priori unknown parameters to perform various maneuvers and trajectory control motion with complex constraints. Proposed iterative algorithm modification that speeds up the learning process for long trajectories.

LEARNING. MOTION CONTROL. CONSTRAINTS. ADAPTATION. MOBILE ROBOT. OPTIMIZATION. COVARIANCE MATRIX ADAPTATION STABILITY. ADAPTATION.

Развитие вычислительной техники и портативных источников энергии обеспечило благоприятные условия для развития автономных малогабаритных мобильных роботов, решающих задачи военного, научного и промышленного значения. При этом во многих задачах, таких, как патрулирование и разведка территории, транспортировка грузов необходимо точное следование по заранее определенному маршруту.

Аналитическое решение задачи синтеза и оптимизации траекторного управления мобильным роботом требует знания большого количества параметров как самого робота, так и среды, в которой он находится. Для оптимального управления и обхода препятствий необходимо учитывать не только кинематические ограничения, связанные с геометрическими свойствами робота и объектов окружающей среды, но и динамические, вызванные тем, что в материальном мире действуют законы сохранения, а объекты обладают массой, импульсом, энергией и к ним приложены различные внешние силы.

При использовании мобильных роботов в реальной среде для решения практических задач данные, необходимые для построения модели, часто отсутствуют в силу невозможности их точного измерения, либо в случае динамического окружения, когда параметры среды меняются со временем и под воздействием различных процессов. Так, в процессе эксплуатации механического робота, наряду с запаздыванием по времени исполнительных устройств, могут происходить изменения его массово-инерционных параметров, например, при добавлении или сбросе груза, а также изменения характеристик самих исполнительных устройств – изменение давления в шинах и, как следствие, изменение реакции объекта управления на команды от управляющей системы. Следовательно, для корректного и эффективного выполнения поставленных задач необходима оперативная корректировка управляющих действий на основании данных, полученных в ходе работы.

Примерами задач, требующих адаптивного

управления, являются: движение колесного робота по некоторой траектории с использованием поперечного скольжения колес для выполнения маневров на большой скорости, например, таких, как поворот под прямым углом или «полицейский» разворот; управление беспилотным летательным аппаратом, например, вертолетом в процессе движения по определенной траектории с заданными значениями скоростей и ориентацией в пространстве.

Существующие работы

Актуальность задачи синтеза траекторного управления подтверждается большим количеством подходов и научных работ [1–3]. Однако в большинстве подобных работ закон управления строится на основании динамических моделей, предполагающих точное знание динамических характеристик системы, что делает их неприменимыми для управления объектом с частично известными параметрами в плохо формализованной среде.

Для реализации адаптивного управления и решения проблемы неполноты знаний о системе используются искусственные нейронные сети или нейроноподобные системы управления [4–7]. С их помощью строятся нелинейные кинематические и динамические модели объекта управления, либо адаптируются и уточняются модели, построенные аналитически. Так, в [4] используется персептрон с одним скрытым слоем для адаптации эталонной модели динамики к реальным управляющим устройствам летательного аппарата. В работах [5, 6] нейроноподобная система управления используется для распознавания образов состояний, в которых оказался объект управления, и для сопоставления этим образам одного из заранее определенных действий, выполнение которого улучшает «эмоциональную» (качественную) оценку состояния робота.

У нейронных сетей есть ряд существенных недостатков, ограничивающих их широкое применение в системах управления [7]: эффект забывания, возникающий при попытке дообучить обученную ранее сеть; сложность формирования обучающей выборки; длительное обучение, требующее значительных вычислительных ресурсов; проблема создания динамических нейронных сетей.

Рассматриваемую задачу траекторного управления можно сформулировать как задачу нахож-

дения управляющего вектора, минимизирующего отклонение положения робота в фазовом пространстве относительно заданной траектории, другими словами – нахождение минимума штрафной функции. Для большинства реальных задач штрафные функции являются нелинейными, невыпуклыми, плохо обусловленными и высокой размерности $N > 10$, что делает применение методов, основанных на производных, невозможным, либо неэффективным, особенно в случае отсутствия динамической модели робота или при наличии дополнительных ограничений на решение. Среди методов оптимизации без использования производных одним из наиболее стабильных и эффективных является алгоритм адаптации ковариационной матрицы CMA-ES (Covariance Matrix Adaptation Evolution Strategy) [8, 9] и его модификация Basin-CMA [10], позволяющая учитывать ограничения, наложенные на решение задачи.

В данной статье рассматривается применение алгоритма Basin-CMA для решения задачи траекторного управления автономным мобильным роботом и его обучения выполнению различных маневров, а также предложен подход, увеличивающий скорость обучения.

Математическая модель

Без ограничения общности рассматривается плоское движение четырехколесного робота, состояние которого описывается координатами в фазовом пространстве $\vec{s} = (x, y, \theta, \dot{x}, \dot{y}, \dot{\theta})^T$, где x и y – декартовы координаты некоторой точки робота C (рис. 1), например, геометрического центра или центра одной из колесных осей, а θ – ориентация робота, угол между продольной осью робота и положительным направлением оси OY .

Управление роботом дискретно и осуществ-

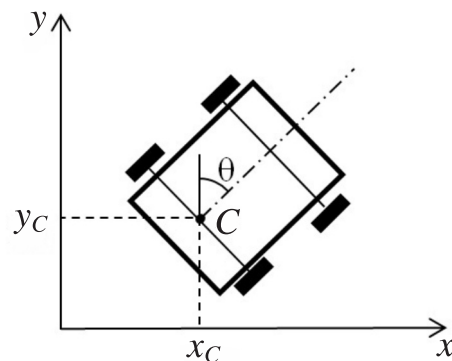


Рис. 1. Координаты робота

вляется с фиксированным временным квантом управления Δt . На каждом такте управления робот выполняет действия, описываемые вектором $\vec{d} = (d_1, d_2, \dots, d_M)^T$, где M – количество исполнительных устройств. Для каждого исполнительного устройства определено линейно упорядоченное множество допустимых действий $D_i = \{d_{i1}, d_{i2}, \dots, d_{ik_i}\}$, $i = 1, M$, где k_i – количество возможных действий, а порядок элементов определяется величиной воздействия на объект управления. Целевая траектория задается в виде множества точек $T = \{s_0, s_1, \dots, s_F\}$ в фазовом пространстве, описывающих, в каком состоянии должен находиться робот в заданные моменты времени, s_0 – начальное состояние, s_F – конечное.

Рассматривается задача нахождения вектора управления $\vec{u} = (\vec{d}_1, \vec{d}_2, \dots, \vec{d}_N)^T$, минимизирующего отклонение реализуемой траектории робота относительно заданной. Здесь N – количество тактов управления, зависящее от длины траектории (времени, отведенного на ее прохождение) и длительности временного кванта между тактами управления.

В разных задачах отклонение может определяться и рассчитываться различными способами. В данной статье, без ограничения общности, в качестве отклонения эмпирической траектории от целевой рассматривается среднее квадратическое отклонений по евклидовой норме реальных фазовых координат робота относительно заданных.

Задача определения координат робота в настоящей статье не рассматривается. Предполагается, что робот оснащен всеми необходимыми сенсорами для определения его положения, ориентации в пространстве и скорости перемещения.

Описание алгоритма Basin-CMA

Алгоритм адаптации ковариационной матрицы с бассейном (Basin-CMA) [10] создан на базе алгоритма CMA [8]. Его главным отличием от предшественника является возможность учитывать дополнительные ограничения, наложенные на решение задачи, например, такие, как дискретность множества решений и соблюдение кинематических и динамических ограничений при движении робота. Алгоритм адаптации ковариационной матрицы разработан Хансеном [8], как эволюционная стратегия для адаптации мутационного распределения на протяжении смены поколений. Каждая его итерация состоит из трех шагов.

1. Генерация популяции из λ нормально распределенных векторов размерности n :

$$\vec{x}_i^{(g+1)} = \vec{m}^{(g)} + \sigma \vec{z}_i, \quad \vec{z}_i \in N(0, C^{(g)}), \quad i = \overline{1, \lambda},$$

где $g+1$ – номер текущей итерации; $\vec{m}^{(g)} \in \mathbb{R}^n$ – вектор математического ожидания; $\sigma \in \mathbb{R}_+$ – размер шага – коэффициент, управляющий масштабом разброса векторов в популяции; $C^{(g)} \in \mathbb{R}^{n \times n}$ – симметричная, положительно определенная матрица ковариаций, полученная на итерации g .

2. «Селекция и мутация». Для каждого из полученных векторов $\vec{x}_1^{(g+1)}, \vec{x}_2^{(g+1)}, \dots, \vec{x}_\lambda^{(g+1)}$ вычисляется значение целевой функции f и выбираются $\mu < \lambda$ «элитных» векторов, наилучшим образом минимизирующих целевую функцию, т. е. вектора, на которых целевая функция принимает наименьшие значения по сравнению с остальными. Элитные вектора упорядочиваются в порядке возрастания значений функции f :

$$f(\vec{x}_{1:\lambda}^{(g+1)}) \leq f(\vec{x}_{2:\lambda}^{(g+1)}) \leq \dots \leq f(\vec{x}_{\mu:\lambda}^{(g+1)}).$$

С каждым элитным вектором связывается вес:

$$\omega_j = \frac{\ln(\mu + 1) - \ln j}{\mu \ln \mu - \ln \mu!}, \quad j = \overline{1, \mu}.$$

Веса выбираются таким образом, чтобы больший вес был у вектора, для которого значение целевой функции минимально. Возможно использование и других схем расчета весов при выполнении условий $\omega_1 \geq \omega_2 \geq \dots \geq \omega_\mu > 0$ и $\sum_{k=1}^{\mu} \omega_k = 1$.

Используя найденные элитные вектора и их веса, вычисляется новый вектор математического ожидания:

$$\vec{m}^{(g+1)} = \sum_{j=1}^{\mu} \omega_j \vec{x}_{j:\lambda}^{(g+1)}.$$

3. Адаптация ковариационной матрицы в соответствии с полученными результатами:

$$C^{(g+1)} = (1 - c_{\text{cov}})C^{(g)} + c_{\text{cov}} \sum_{j=1}^{\mu} \omega_j (\vec{x}_{j:\lambda}^{(g+1)} - \vec{m}^{(g)})(\vec{x}_{j:\lambda}^{(g+1)} - \vec{m}^{(g)})^T,$$

где $c_{\text{cov}} \in (0, 1)$ – параметр, задающий скорость обучения.

Более подробно алгоритм CMA-ES описан в работах [8, 9]. Выполнение алгоритма продолжается до тех пор, пока не будет найдено оптимальное решение или не будет достигнут наперед

заданный максимум количества итераций. В ходе работы алгоритма математическое ожидание нормального распределения стремится к глобальному минимуму функции.

В Basin-CMA помимо основных параметров алгоритма CMA и целевой функции f задается множество ограничений вида $\phi_i(x) = 0 : i \in \overline{1, l}$ и $\psi_j(x) \leq 0 : j \in \overline{l+1, m}$. Каждый вектор $\bar{x}_i, i = \overline{1, \lambda}$, сгенерированный на первом этапе, проецируется на эти ограничения, полученные проекции используются на втором и третьем шаге алгоритма. Под проекцией подразумевается ближайший по евклидовой норме вектор, удовлетворяющий всем ограничениям.

Несмотря на высокую скорость сходимости метода Basin-CMA, его применение в чистом виде для задач высокой размерности с эмпирическим вычислением целевой функции является ресурсоемким. Например, для обучения двухколесного робота движению по криволинейной траектории в форме синусоиды в течение 50 с при длине временного кванта управления 0,5 с и двух исполнительных устройствах, размерность искомого вектора составляет 200 элементов. Поиск вектора управления, реализующего движение по описанной траектории с максимальным отклонением по координатам, не превышающем 0,1 м при движении со скоростью 1 м/с, требует примерно 350 итераций, на каждой из которых целевая функция вычисляется по 19 раз.

Такое количество итераций приемлемо для компьютерной модели и числовых расчетов, но является большим для механического робота в реальных условиях, когда «вычисление» значения целевой функции для заданного вектора управления является применением этого вектора для управления в процессе движения и фиксации состояний робота в заданные моменты времени. Подобное эмпирическое вычисление требует значительного количества ресурсов. Следовательно, для решения практических задач необходимо снижение количества итераций.

Итеративный подход

В основе предлагаемого подхода лежит разбиение исходной траектории на небольшие отрезки, для каждого из которых размерность вектора управления $10 \leq n \leq 20$. Методом Basin-CMA ищется управление для первого отрезка. Затем, используя полученное решение, ищется вектор управления для участка траектории, соответствующего

первым двум отрезкам, и т. д. до тех пор, пока не будет получен вектор управления заданной длины и реализующий целевое поведение. Рассмотрим алгоритм поиска более подробно.

Начальный вектор управления $\bar{u} = (\bar{d}_1, \bar{d}_2, \dots, \bar{d}_N)^T$ разбивается на $k = \left\lfloor \frac{N}{p} \right\rfloor$ частей малой размерности $p \in [5, 10]$:

$$\bar{u}_1 = (\bar{d}_1, \bar{d}_2, \dots, \bar{d}_p)^T, \bar{u}_2 = (\bar{d}_{p+1}, \bar{d}_{p+2}, \dots, \bar{d}_{2p})^T, \dots, \bar{u}_k = (\bar{d}_{(k-1)p+1}, \bar{d}_{(k-1)p+2}, \dots, \bar{d}_N)^T.$$

На первом шаге ищется управление \tilde{u}_1 для участка траектории, соответствующего первым p тактам управления. Для этого методом Basin-CMA решается задача оптимизации штрафной функции размерности pM , где M – количество исполнительных устройств. В качестве начального вектора математического ожидания используется вектор \bar{u}_1 и единичная матрица в роли матрицы ковариаций.

На втором шаге рассматривается участок траектории, соответствующий первым $2p$ тактам управления. А в методе оптимизации в роли начального решения используется вектор $\bar{m} = \tilde{u}_1 \cup \bar{u}_2$, где \tilde{u}_1 – решение, полученное на первом шаге. Матрица ковариаций $C_2 = \{c_{ij}\} \in \mathbb{R}^{2pM \times 2pM}$, где

$$c_{ij} = \begin{cases} 0, 01, & i = j, i = \overline{1, pM} \\ 1, & i = j, i = \overline{pM+1, 2pM} \\ 0, & i \neq j \end{cases}$$

При таком выборе ковариационной матрицы C на первых итерациях алгоритма Basin-CMA последние pM компонент вектора математического ожидания варьируются в 10 раз сильнее, чем компоненты вектора, полученные на предыдущем шаге. В первую очередь осуществляется поиск управления для добавленного участка траектории с одновременной оптимизацией и адаптацией управления, найденного ранее.

Таким образом, на шаге k решается задача поиска оптимального управления для участка, состоящего из первых k частей траектории. Начальный вектор математического ожидания $\bar{m}_k = \tilde{u}_{k-1} \cup \bar{u}_k$ и матрица ковариаций $C_k = \{c_{ij}\}$, где

$$c_{ij} = \begin{cases} 0, 01, & i = j, i = \overline{1, (k-1)pM} \\ 1, & i = j, i = \overline{(k-1)pM+1, kpM} \\ 0, & i \neq j \end{cases}$$

Результатом выполнения последнего шага $k = \left\lfloor \frac{N}{p} \right\rfloor$ является искомый вектор управления \tilde{u} .

Экспериментальные исследования

Исследование применимости и эффективности предложенных методов синтеза управления произведено на реальном четырехколесном мобильном роботе с двумя каналами управления скоростью вращения колес, реализующими управление движением робота, аналогичное управлению с классическим «дифференциальным» приводом. Множество допустимых действий каждого из исполнительных устройств $D_1 = D_2 = \{-10, -9, \dots, 10\}$.

Для решения задачи локализации робота оснащен двумя одометрами, акселерометром, магнетометром и трехосным гироскопом. Комбинация магнетометра и гироскопа позволила определить ориентацию робота с точностью ± 2 градуса. При движении без поперечного проскальзывания точность определения положения робота составляла около 1 % от пройденного пути и около 5 % при движении с продольным и поперечным проскальзыванием колес.

Также для исследований с помощью Microsoft Robotics Developer Studio была создана виртуальная модель робота с дифференциальным приводом. Данная среда моделирования позволяет создавать роботов различной конфигурации, задавать и изменять в процессе движения параметры робота и окружающей среды, например, такие, как масса робота, положение центра масс, а также коэффициенты сил трения продольного и поперечного скольжений. Кроме этого, в Robotics

Studio возможно точное определение положения и ориентации робота в пространстве.

В качестве тестовых траекторий для обучения использовались маневр «полицейский разворот» и движение по криволинейной траектории (синусоиде). Движение по синусоиде оказалось очень удобным для оценки эффективности применения описанных методов на траекториях различной длины, в силу простоты описания траектории и ее масштабируемости. В роли целевой функции выбрано среднеквадратическое отклонение фазовых координат робота в процессе движения относительно целевой траектории, в заданные моменты времени.

И в реальной, и в виртуальной среде использование предложенной модификации алгоритма Basin-CMA позволило быстро находить вектор управления для движения по заданному маршруту и адаптироваться к изменениям параметров робота.

При изучении предложенного итеративного подхода рассматривалось разбиение исходного вектора управления на части, соответствующие $p = 5$ тактам управления и размерности $n = 10$. Использование меньшей размерности для подзадач лишь увеличивало их количество, а число итераций, необходимое для их решения, оставалось без изменения. Среднее количество учебных заездов на решение одной подзадачи (выполнение одного шага) составляло около $300 \div 350$. Как видно из рис. 2, решение задач большей размерности требует существенно большего количества итераций.

На рисунке показано, что по количеству «вычислений» целевой функции итеративный подход дает незначительное преимущество по сравнению

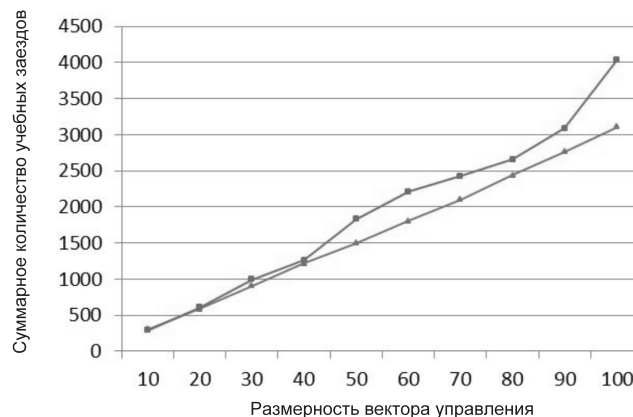


Рис. 2. Суммарное количество заездов во время обучения, в зависимости от длины траектории (размерности исходной задачи) (— \blacktriangle —) итеративный подход; (— \blacksquare —) Basin-CMA

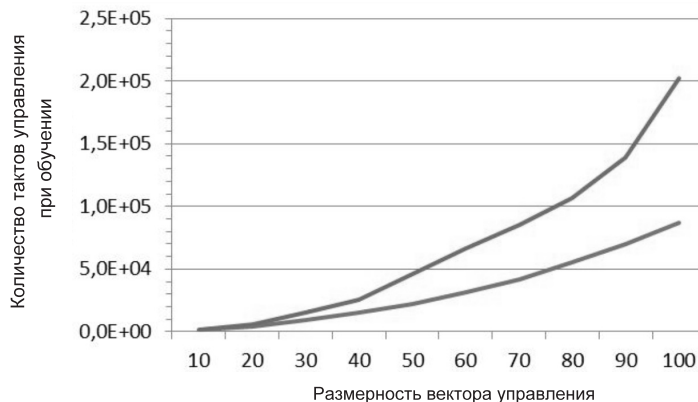


Рис. 3. Зависимость времени обучения от размерности исходной задачи (—) итеративный подход; (—) Basin-CMA

с базовым алгоритмом. Однако в данном сравнении важно учесть, что в случае использования оригинального Basin-CMA для решения исходной задачи, на каждой итерации робот осуществляет движение по всей длине траектории, а в предложенном подходе – только по ее части. Для пути, пройденного в процессе обучения и измеренного в тактах управления, получены данные, представленные на рис. 3.

Таким образом, для задачи с исходной размерностью $n_0 = 100$ применение предложенного итеративного подхода с разбиением вектора управления на $k = 10$ частей равной размерности дало выигрыш в два раза по времени обучения и пройденному пути, по сравнению с обучением с использованием базового метода Basin-CMA.

Проведенные исследования показали, что метод Basin-CMA применим для автономного обучения мобильного робота выполнению различных маневров и синтеза траекторного управления, а также для оптимизации существующих программных управлений. Базовый и модифицированные методы работают в условиях невозможности построения точной математической модели, а системы управления, построенные на их основе, способны адаптироваться к нелинейным и изменчивым свойствам объекта и к изменениям в окружающей среде. Предложенный подход существенно ускоряет поиск оптимального управления для протяженных траекторий, требующих большого количества управляющих действий.

СПИСОК ЛИТЕРАТУРЫ

1. Бурдаков С.Ф. Системы управления движением колесных роботов [Текст] / С.Ф. Бурдаков, И.В. Мирошник, Р.Э. Стельмаков. –СПб.: Наука, 2001.
2. Stotsky, A. Sliding mode control of a car like mobile robot using single track dynamic model [Text] / A. Stotsky, X. Hu, M. Egerstedt // Proc. of the 14th World Congress of the IFAC. –1999. –Vol. B. –P. 119–124.
3. Мирошник, И.В. Нелинейное и адаптивное управление сложными динамическими системами [Текст] / И.В. Мирошник, В.О. Никифоров, А.Л. Фрадков. –СПб.: Наука, 2000.
4. Johnson, E. Adaptive trajectory control for autonomous helicopters [Text] / E. Johnson, S. Kannan // J. of Guidance, Control, and Dynamics. –2005. –Vol. 28. –P. 524–538.
5. Жданов, А.А. Нейронная адаптивная система управления [Текст] / А.А. Жданов, М.В. Крыжановский // V Всерос. науч.-техн. конф. Нейроинформатика-2003. –2003. –С. 163–168.
6. Жданов, А.А. Автономный искусственный интеллект [Текст] / А.А. Жданов. –М.: БИНОМ. Лаборатория знаний, 2008.
7. Чернодуб, А.Н. Обзор методов нейроуправления [Текст] / А.Н. Чернодуб, Д.А. Дзюба // Проблемы программирования. –2011. –№ 2. –С. 79–94.
8. Hansen, N. Adapting arbitrary normal mutation distributions in evolution strategies: The covariance matrix adaptation [Text] / N. Hansen, A. Ostermeier // Proc. of the 1996 IEEE Conf. on Evolutionary Computation. –1996. –P. 312–317.
9. Auger, A. Experimental Comparisons of Derivative Free Optimization Algorithms [Text] / A. Auger, N. Hansen, J. Perez Zerpa [et al.] // Proc. of the 8th International Symp. on Experimental Algorithms. –2009. –P. 3–15.
10. Wampler, K. Optimal Gait and Form for Animal Locomotion [Text] / K. Wampler, Z. Popović // ACM Transactions on Graphics –2009. –Vol. 28. –№ 3.