УДК 658.62.018.012 DOI 10.21685/2307-4205-2020-3-10

А. Н. Дарьина, И. В. Прокопьев

МЕТОД ОПТИМИЗАЦИИ ПАРАМЕТРОВ КОНТРОЛЛЕРА БЕСПИЛОТНОГО ТРАНСПОРТНОГО СРЕДСТВА НА ОСНОВЕ ОПТИМИЗАЦИИ РОЯ ЧАСТИЦ

A. N. Daryina, I. V. Prokopiev

AN OPTIMIZATION METHOD OF THE UNMANNED VEHICLE CONTROLLER'S PARAMETERS BASED ON THE OPTIMIZATION OF PARTICLES'S ROY

Аннотация. Для беспилотного транспортного средства в сложных условиях, когда пространственные ограничения серьезно сужают пространство допустимых состояний, стратегия выбора пространства состояний более эффективна, чем выборка в пространстве управлений. Хотя это было очевидно, практический вопрос заключается в том, как его достичь, одновременно удовлетворяя жестким ограничениям динамической осуществимости транспортного средства. В этой статье представлена система управления беспилотного транспортного средства на основе контроллера модели прогнозирующего интегрального пути (MPPI), глубокой сверточной нейронной сети (CNN) для понимания сцены в реальном времени и метода оптимизации роя частиц (PSO) для нахождения вектора оптимальных параметров функции стоимости. Метод основан на оптимизации функции стоимости, которая определяет, где на поверхности пути должно двигаться транспортное средство.

Ключевые слова: беспилотное транспортное средство, модель прогностического управления, нейронная сеть, метод оптимизации роя частиц.

Abstract. For an unmanned vehicle, in difficult conditions, when spatial constraints seriously narrow the space of admissible states, the strategy of choosing a state space is more effective than sampling in the control space. Although this was obvious, the practical question is how to achieve it while meeting the stringent constraints of the vehicle's dynamic feasibility. This article presents an unmanned vehicle control system based on the predictive integrated path model (MPPI) controller, deep convolutional neural network (CNN) for real-time scene understanding and particle swarm optimization (PSO) to find the vector of optimal cost function parameters. The method is based on the optimization of the cost function, which determines where the vehicle should move on the surface of the path.

Keywords: unmanned vehicle, predictive control model, neural network, particle swarm optimization method.

Введение

Целью управления беспилотного транспортного средства (БТС) обычно является перевод БТС из начального состояния в заданное конечное состояние или осуществление (отслеживание) заданного программного движения БТС. Синтезируемые законы управления должны обеспечивать требуемые показатели качества (точность, быстродействие и т.п.) по всем управляемым координатам с учетом заданных ограничений на управление и состояние БТС. Кроме того, практически важно, чтобы эти законы управления были оптимальными по отношению к заданному функционалу качества. Однако в ряде случаев этого недостаточно и требуется синтезировать законы управления, обеспечивающие достижение цели управления в широком классе неопределенности модели динамики БТС. С другой стороны, система управления должна обеспечить управление с учетом динамических ограничений в реальном времени, параметры которых заранее не известны.

Сложность решения рассматриваемой задачи классическими методами (метод Понтрягина, метод редукции к задаче нелинейного программирования и другие эвристические методы) заключается в том, что требуется большое количество вычислений и он не может быть реализован на борту робота в процессе его эксплуатации.

© Дарьина А. Н., Прокопьев И. В., 2020

Наиболее общий подход к решению задач управления в реальном времени основан на использовании метода предсказания траектории, обеспечивающей возможность добиться прогресса на пути достижения цели [1]. Основная проблема реализации этого подхода состоит в том, что при моделировании необходимо знать не только информацию о модели, но и информацию об окружающей среде. В работе [2] предлагается генерировать множество траекторий с различным управлением и выбирать их по дополнительному функционалу, который учитывает внешние факторы, в том числе и динамические фазовые ограничения. Данный функционал определяется разработчиком системы управления и ставится на борт объекта до реализации алгоритма управления. Построение такой функции требует существенного предварительного анализа внешней среды и сопряжено с возможностью внесения корректировки при выборе траектории.

Тем не менее все эти подходы основаны на очень точной позиции из внешнего источника (либо GPS, либо захват движения). Рассматривается автономное вождение в сложных условиях с активным вождением, используя только внутренние датчики, такие как камеры и ИДУ. Есть несколько способов решения этой проблемы. Существует много подходов SLAM, в которых для обеспечения точного положения используются камеры [3, 4], LIDAR [5] или другие комбинации датчиков [6]. Эти системы обычно обеспечивают положение относительно сгенерированной карты. Эти методы имеют тенденцию быть вычислительно дорогими. Альтернативный метод обеспечения абсолютной позиции использует глубокие нейронные сети для прямой регрессии позиции оценки в районе, посещенном ранее [7]. Однако этот метод локализации еще недостаточно точен, чтобы его можно было напрямую использовать для контроля.

В данной работе применяется модель прогностического интегрального управления (MPPI) [8, 9]. Метод основан на оптимизации функции стоимости, которая определяет, где на поверхности пути должно двигаться транспортному средству. Поэтому поверхность должна кодировать текущее и будущее положение дороги, препятствий, пешеходов и другие транспортные средства. Сеть обучена так, чтобы стоимость была самая низкая в центре дорожки и выше от центра. Эта карта стоимости может затем непосредственно вводиться в алгоритм прогнозного управления моделью. Прогнозное управление моделью работает путем чередования оптимизации и выполнения: сначала оптимизируется последовательность управления с разомкнутым контуром (генерируются тысячи траекторий после чего считается их стоимость на основе функционала, конкретный вид и параметры которого зависят от ручной настройки и личного опыта разработчика), затем первое управление в этой последовательности выполняется транспортным средством и затем принимается обратная связь о состоянии, и весь процесс повторяется.

Параметры оператора стоимости в алгоритме MPPI настраиваются вручную, а перспективным подходом является применение методов обучения с подкреплением (RL). Большинство предыдущих работ с MPC фокусируется на задачах стабилизации или отслеживания траектории. Ключевой разницей между классическими MPC и MPC для обучения с подкреплением является то, что задачи *RL* являются сложными задачами. Сложность целей в задачах обучения с подкреплением заключается в том, что увеличиваются вычислительные затраты на оптимизацию, так как оптимизация должна происходить также в режиме реального времени.

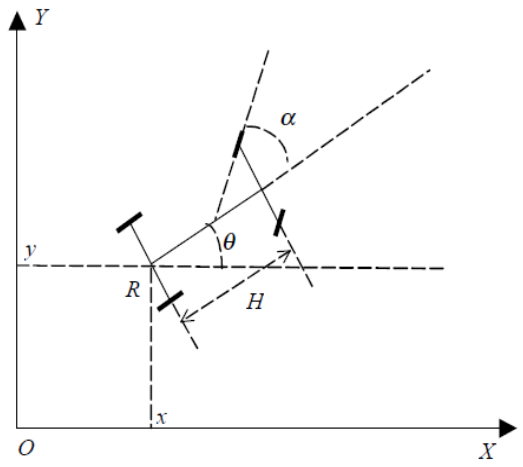
Алгоритм MPPI позволяет в реальном времени обрабатывать сложные нелинейные функции динамики и стоимости (тысячи траекторий), но, как и классические MPC, алгоритм страдает от ухудшения устойчивости, когда моделируемая динамика отличается от истинной динамики транспортного средства. Стратегией борьбы с этой проблемой может быть настройка параметров модели объекта управления и параметров функции стоимости с помощью эволюционного метода обучения с подкреплением, а также применения глубокой сверточной нейронной сети (CNN) для понимания сцены в реальном времени.

Идентификация модели беспилотного транспортного средства

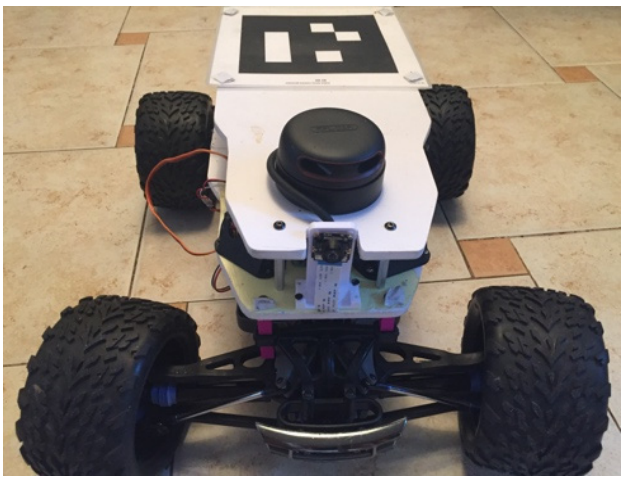
Объектом управления является БТС, построенное с использованием автомобильного шасси 1/10 масштаба (рис. 1,*б*). БТС весит 1,5 кг и способен развивать скорость до 10 м/с.

Вычисления выполняются с помощью комплекта NVIDIA Jetson TX2. Измерение локальных координат движения БТС проводилось в Робототехническом центре ФИЦ ИУ РАН [10].

Два передних колеса БТС служат для рулевого управления, а два задних колеса – для отслеживания. Кинематическая схема управления движением БТС показана на рис. 1.



а)



б)

Рис. 1. Кинематическая схема управления движением БТС:
α – угол поворота руля; θ – угол ориентации БТС относительно оси x;
H – расстояние между передней и задней осями робота;
R – базовая точка, расположенная по середине задней оси робота; x, y – координаты

Задача идентификации обычно формулируется как задача оценивания параметров модели системы, которая обладает существенными чертами реальной системы и характеризует ее динамические свойства в удобной для синтеза управления форме.

Традиционным методом построения моделей динамических объектов является реализация процедуры идентификации с использованием регрессионных модельных структур.

Регрессия непосредственно связана с задачей прогнозирования величины выхода $\mathbf{y}(t)$ на основе информации, полученной при измерении других величин φ_i, $i=\overline{1,d}$, содержащих информацию о прошлом поведении системы.

Для БТС, изображенного на рис. 1,*б*, задача идентификации состоит в нахождении функции $g(\varphi)$ такой, чтобы оценка $\hat{y}=g(\varphi)$ удовлетворяла некоторому критерию и была прогнозом величины $\mathbf{y}(t)$. Функция $g(\varphi)$ – функция регрессии, должна быть некоторым методом оценена по экспериментальным данным, т.е. $g(\varphi)=g(\varphi,\mathbf{q})$, где $\mathbf{q}$ – вектор настраиваемых параметров.

Задача состоит в сборе необходимого количества экспериментальных данных во всем рабочем диапазоне системы $Z^L=\{\mathbf{u}(t),\mathbf{y}(t)\}$, $t=\overline{1,L}$. Полнота и достоверность полученных данных во многом определяют качество идентификации.

Таким образом, динамический объект может быть представлен в следующем виде:

$$\hat{y}(t\,|\,\mathbf{q})=g(\varphi(t,\mathbf{q}),\mathbf{q}). \tag{1}$$

Прогнозирующая модель динамики модели мобильного робота $g(\varphi(k,\mathbf{q}),\mathbf{q})$ может быть реализована на нейросетевой модельной структуре, имеющей следующее математическое представление:

$$g_i(\varphi(t,\mathbf{q}),\mathbf{q})=\hat{y}_i(t\,|\mathbf{q})=\hat{y}_i(t\,|w,W)=F_i\left(\sum_{j=1}^{n_h}W_{ij}f_j\left(\sum_{l=1}^{n_\varphi}w_{jl}\varphi_l+w_{j0}\right)+W_{i0}\right), \tag{2}$$

где $f_j(x)=\tanh(x)$ – активационная функция нейронов скрытого слоя; $F_i(x)=ax$; $a=\text{const}$ – активационная функция нейронов выходного слоя; n_φ – размерность регрессионного вектора (число входов ИНС); n_h – число нейронов в скрытом слое; $\mathbf{q}$ – вектор настраиваемых параметров нейронной сети, включающий весовые коэффициенты и нейронные смещения (w_{jl},W_{ij}).

При использовании выражения (2) оптимизация параметров **q** представляет собой отображение множества экспериментальных данных на множество параметров модели (1). Традиционно используемым критерием оптимальности является среднеквадратичная ошибка прогнозирования

$$\sigma_L(\mathbf{q}, Z^L) = \frac{1}{2L}\sum_{k=1}^{L}(y(k) - \hat{y}(k \mid \mathbf{q}))^2 . \quad (3)$$

Таким образом, идентификация параметров модели (1) состоит в нахождении вектора параметров $\mathbf{q}$, минимизирующих критерий (3)

$$\hat{\mathbf{q}} = \arg\min_{\mathbf{q}} \sigma_L(\mathbf{q} Z^L) . \quad (4)$$

Краткий обзор MPPI

Рассмотрим формальную постановку задачи синтеза управления движением по траектории с дискретным временем. Задана математическая модель объекта управления

$$\dot{x}_{t+1} = \mathbf{f}(x_t, v_t), \quad (5)$$

где x_t – вектор состояния объекта, v_t – фактический вектор управления, $x_t \in \mathbf{R}^n$, $v_t \sim N(u_t, q_3)$, $u_t \in \mathbf{U} \subseteq \mathbf{R}^m$, $\mathbf{U}$ – замкнутое ограниченное множество; $m \le n$, $N(u_t, q_3)$ – Гауссов шум с математическим ожиданием u_t. Управление должно пройти через контроллер нижнего уровня со стохастической добавкой в виде Гауссова шума, в результате фактический вектор управления определим как $V = (v_0, v_1 .... v_{T-1})$ – последовательность входов через некоторое количество шагов *T*. Нас интересует вычисление среднего управления $U = (u_0, u_1 ... u_{T-1})$, которое минимизирует функцию стоимости

$$J(U) = \left[q_1(x_T) + \sum_{t=0}^{T-1} \left[q_2(x_t) + q_3 u_t^T q_4^{-1} u_t \right] \right], \quad (6)$$

где $q_1(\cdot)$ – терминальная стоимость состояния, а $q_2(\cdot)$ – мгновенная стоимость состояния, $S(V^m) = q_1(x_T^m) + \sum_{t=0}^{T-1} q_2(x_t^m)$ – часть стоимости, зависящая от состояния.

Алгоритм MPPI начинается с инициализации вектора состояния x_0. Вычисление управления начинается с учета управляющей последовательности из предыдущей итерации U_{k-1}, симуляции (тысячи) траектории параллельно, каждая с различной последовательностью управления V^m, где *m* – реализация случайной траектории.

Затраты на управление собираются для каждого развертывания и сопоставлены с весами траектории:

$$w(V^m) = \exp\left(-\frac{1}{q_2} \left(S(V^m) - \sum_{t=0}^{T-1} u_{t,k-1}^T q_3^{-1} v_t^m - \rho \right) \right), \quad (7)$$

ρ – устанавливается в минимальное значение затрат среди всех выбранных траекторий, предназначен для арифметического недопущения. Квазиоптимальное управление на каждом временном шаге вычисляется как

$$u_{t,k} = u_{t,k-1} + \frac{1}{\sum_{m=1}^{M} w(V^m)} \sum_{m=1}^{M} \left[w(V^m) \xi^m \right], \quad (8)$$

где $\xi^m \sim N(0, q_3)$ – Гауссов шум с нулевым средним; *M* – число реализаций траекторий; $\mathbf{q}$ – вектор идентифицируемых параметров функции стоимости, $\mathbf{q} = [q_1, q_2, q_3, q_4]^T$.

Оптимизация параметров функции стоимости

В настоящей работе рассматривается подход, при котором параметры $\mathbf{q}$ в (6) и нахождения управления (8) определяются методом оптимизации роя частиц (PSO) [11, 12].

В общем случае для нелинейной модели объекта (5) и функционала произвольного вида не известны универсальные методы построения синтезирующей функции (8) даже для решения частной задачи синтеза. Заметим также, что синтезирующая функция (8) может явно не зависеть от начальных значений [13]. Тем более нет универсальных методов для нахождения параметров функции стоимости. Тогда с помощью вычислительной машины можно организовать поиск решения $\tilde{u}(\cdot)$ на множестве параметров $\mathbf{q}$ по дополнительному функционалу в виде определения терминальных условий.

В реальности основной задачей регулирования является максимальное приближение выхода объекта к желаемому значению, определяемому уставкой. Таким образом, более естественно выглядит критерий типа

$$J(\mathbf{q}) = \sum_{k=1}^{N} \left(\left\| \mathbf{x}(t_f) - r(t) \right\| \right)_{\mathbf{x}(0)=\mathbf{x}^{0,k}} \to \min, \tag{9}$$

где $r(t)$ – уставка, траектория для численного синтеза может быть в виде сплайна для дискретного набора точек количеством N; $\mathbf{q}$ – вектор идентифицируемых параметров функции стоимости; $\mathbf{q} = [q_1, q_2, q_3, q_4]^T$ – вычисленные для оптимального управления $\tilde{u}(\cdot)$, полученного методом PSO, используя следующие процедуры:

$$q_{k+1}^i = q_k^i + v_{k+1}^i, v_{k+1}^i = w_k v_k^i + c_1 r_1 (p_k^i - q_k^i) + c_2 r_2 (p_k^g - q_k^i), \tag{10}$$

где q_k^i – величина параметра; k – текущая итерация; v_k^i – величина и направление вектора скорости каждого параметра; w_k – постоянный вес; c_1, c_2 – параметры ускорения; r_1, r_2 – случайные числа от 0 до 1; p_k^i – лучшая найденная точка варианта параметра; p_k^g – лучшая найденная точка всех вариантов параметра.

После вычисления направления вектора v параметр перемещается в точку q_{k+1}^i. В случае необходимости обновляются значения лучших точек для каждого параметра и для всех параметров в целом. После этого цикл повторяется. Так как p_k^i, p_k^g находятся после вычисления (9) (движение БТС по траектории), то алгоритм требует значительных ресурсов и запускается десятки тысяч итераций на симуляторе.

Для сужения области поиска используется принцип базисного решения. Согласно этому принципу [13] мы определяем начальные значения параметров, которые называем базисными. Тогда поиск решения сосредотачиваем в окрестности базисного решения. Если базисное решение выбрано удачно, то время поиска решения можно существенно сократить. В любом случае базисное решение можно всегда заменить хорошим решением, найденным в процессе поиска. Принцип поиска решения на основе базисного удобен при решении практических задач синтеза управления, где структуру управления может задать инженер, руководствуясь здравым смыслом и опытом.

Сверточная нейронная сеть (CNN)

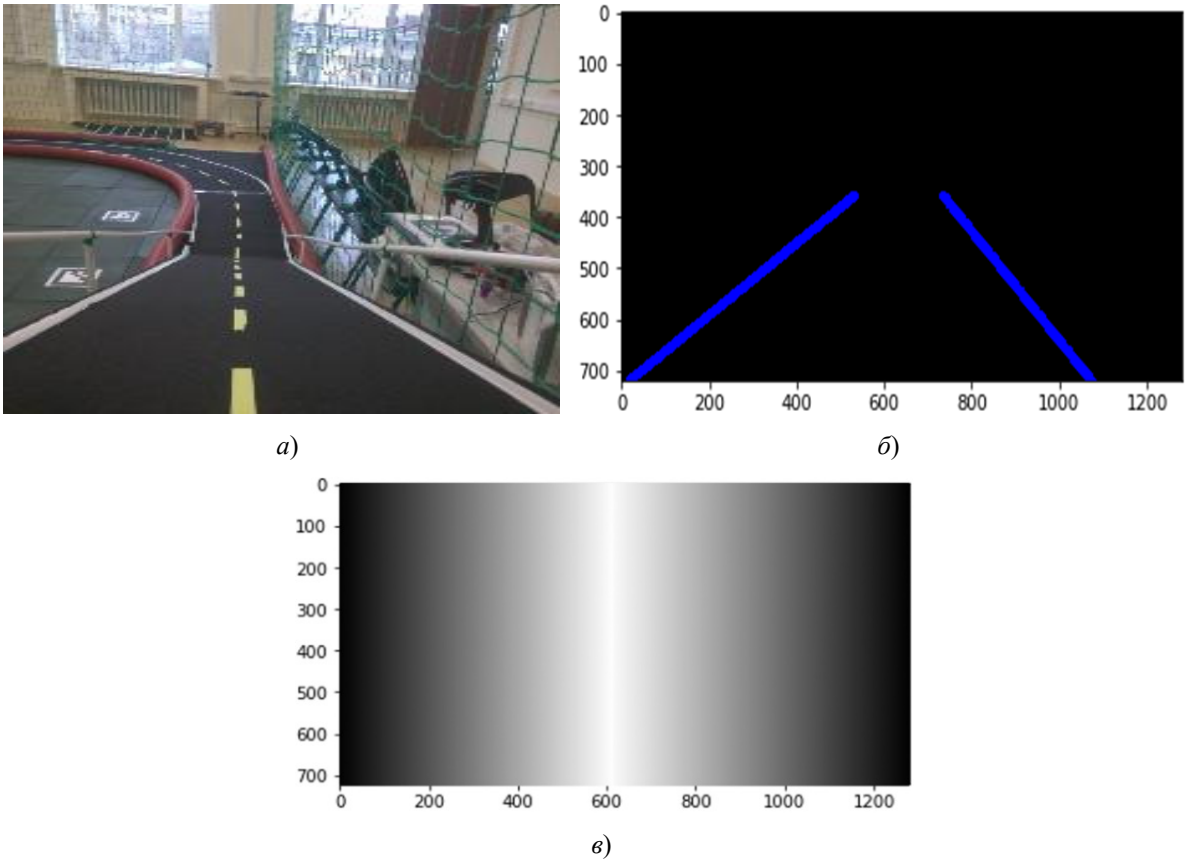
Для того, чтобы моделируемая динамика не отличалась от истинной динамики транспортного средства, необходимо точно вычислять свое состояние относительно затрат на пути к цели в прогнозируемом пространстве. На прогнозируемом пространстве в статье [14] предлагается использовать монокулярное зрение и глубокую сверточную нейронную сеть (CNN) для преобразования сцены в карту затрат в реальном времени.

Чтобы изучить функцию регрессии по пикселям, способную производить затраты на прохождение для каждого пикселя, данные обучения нужны порядка сотен тысяч кадров. Маркировка всех этих данных вручную – трудоемкий, медленный и склонный к ошибкам процесс. Необходимы датчики и камеры, которые могут связывать каждое изображение с полной оценкой состояния, включая

ориентацию и положение в сочетании с обзорной картой трека, зарегистрированного по координатам GPS, они могут быть использованы для создания сотен тысяч маркированных изображений без какой-либо ручной маркировки отдельных картинок.

Наиболее простой подход в создании обучающей маркированной выборки состоит в преобразовании реальной сцены с одной камеры с помощью библиотеки OpenCV в плотную карту затрат. На первом этапе находятся ключевые точки, выделяются ограничения трека (рис. 2,*б*). На втором этапе необходимо произвести трансформацию изображения с помощью матрицы гомографии по четырем или более точкам. На третьем этапе происходит кодировка изображения. Ограничения кодируются числом 255, а предпочтительный путь – нулем. Пиксели от 0 до 255 изменяются по градиенту. Сгенерированная методом MPPI траектория будет стоить меньше, если будет условно проходить по пикселям с меньшим числом.

Кроме того, подъемы и спуски могут кодироваться определенным числом в зависимости от крутизны.



а) *б*)

в)

Рис. 2. Преобразование реальной сцены с одной камеры в плотную карту затрат

С помощью камеры БТС (см. рис. 1), которую мы использовали в наших экспериментах, были сохранены 15 000 последовательных изображений размером 160×120 пикселей. С помощью программы преобразования были получены соответствующие изображения карты затрат непосредственно перед транспортным средством (в координатах транспортного средства). Это включает в себя значительные изменения в условиях освещения, ограничений и позы БТС на треке. После этого была обучена сверточная нейронная сеть. Архитектура CNN ограничена для работы в режиме реального времени на Nvidia Jetson TX2.

Сеть принимает входные изображения 160×120 и передает их через несколько фильтров свертки и два объединенных слоя, за которыми следует набор из четырех расширенных фильтров сверток. Наши сети используют сверточные блоки 3×3 с BatchNorm и ReLU. По сравнению с прямым методом получения карты затрат с помощью OpenCV, обученная таким образом сеть может терпеть удаление небольших областей трека из-за засвечивания и все еще производить карты стоимости, пригодные для использования.

Заключение

В этой работе представлены полевые эксперименты, демонстрирующие новые возможности синтеза системы управления на основе прогнозирующего контроллера, использующего информацию, в выводе нейронной сети. Карта затрат подходит для планирования, когда точное местоположение на трассе не верно.

Параметры контроллера оптимизируются с помощью эволюционного алгоритма метода оптимизации роя частиц. Моделирование показывает улучшение работы прогнозирующего контроллера MPPI, но зависит от инициализации параметров и выбора критерия оптимизации.

Работа выполнена при частичной поддержке РФФИ (грант № 18-29-03061-мк), результаты раздела 3 получены при поддержке РНФ (проект № 19-11-00258).

Библиографический список

1. *Mayne, D. Q.* Model predictive control: Recent developments and future promise / D. Q. Mayne // Automatica. – 2014. – Vol. 50, № 12. – P. 2967–2986.
2. *Howard, T.* Optimal rough terrain trajectory generation for wheeled mobile robots / T. Howard, A. Kelly // International Journal of Robotics Research. – 2007. – Vol. 26, № 2. – P. 141–166.
3. *Engel, J.* Lsd-slam: Large-scale direct monocular slam / J. Engel, T. Schops, D. Cremers // European Conference on Computer Vision. – Springer, 2014. – P. 834–849.
4. *Mur-Artal, R.* Orb-slam: a versatile and accurate monocular slam system / R. Mur-Artal, J. M. M. Montiel, J. D. Tardos // IEEE Transactions on Robotics. – 2015. – Vol. 31 (5). – P. 1147–1163.
5. *Zhang, J.* Lidar odometry and mapping in real-time / J. Zhang, S. Singh Loam // Robotics : Science and Systems. – 2014. – Vol. 2. – P. 634–637.
6. Kinectfusion: Real-time dense surface mapping and tracking / R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohi, J. Shotton, S. Hodges, A. Fitzgibbon // Mixed and augmented reality (ISMAR), 2011 10th IEEE international symposium. – 2011. – P. 127–136.
7. *Kendall, A.* Modelling uncertainty in deep learning for camera relocalization / A. Kendall, R. Cipolla // Proceedings of the International Conference on Robotics and Automation (ICRA). – 2016. – P. 1–8.
8. Aggressive driving with model predictive path integral control / G. Williams, P. Drews, B. Goldfain, J. M. Rehg, E. A. Theodorou // 2016 IEEE International Conference on Robotics and Automation (ICRA). – 2016. – P. 1433–1440.
9. *Williams, G.* Model predictive path integral control: From theory to parallel computation / G. Williams, A. Aldrich, E. A. Theodorou // Journal of Guidance, Control, and Dynamics. – 2017. – P. 1–14.
10. *Дарьина, А. Н.* Робототехнический центр ФИЦ ИУ РАН / А. Н. Дарьина, А. И. Дивеев, И. В. Прокопьев // Вопросы теории безопасности и устойчивости систем. – Москва : ФИЦ ИУ РАН, 2019. – Вып. 21. – С. 66–77.
11. *Weise, T.* Global Optimization Algorithms – Theory and Application : Ph.D Thesis / T. Weise. – University of Kassel, 2008. – P. 184–186.
12. *Kennedy, J.* Particle swarm optimization / J. Kennedy, R. Eberhart // Proceedings of IEEE International conference on Neural Networks. – 1995. – P. 1942–1948.
13. *Diveev, A.* The Network Operator Method for Search of the Most Suitable Mathematical Equation / A. Diveev, E. Sofronova // Bio-Inspired Computational Algorithms and Their Applications / ed. by Dr. Shangce Gao. – InTech, 2012. – P. 19–42.
14. Information Theoretic MPC for Model-Based Reinforcement Learning Conference / G. Williams, N. Wagener, B. Goldfain, P. Drews, J. M. Rehg, B. Boots, A. E. Theodorou // 2017 IEEE International Conference on Robotics and Automation (ICRA) (29 May – 3 June 2017). – 2017. – P. 361–363.

References

1. Mayne D. Q. *Automatica*. 2014, vol. 50, no. 12, pp. 2967–2986.
2. Howard T., Kelly A. *International Journal of Robotics Research*. 2007, vol. 26, no. 2, pp. 141–166.
3. Engel J., Schops T., Cremers D. *European Conference on Computer Vision*. Springer, 2014, pp. 834–849.
4. Mur-Artal R., Montiel J. M. M., Tardos J. D. *IEEE Transactions on Robotics*. 2015, vol. 31 (5), pp. 1147–1163.
5. Zhang J., S. Singh Loam *Robotics: Science and Systems*. 2014, vol. 2, pp. 634–637.
6. Newcombe R. A., Izadi S., Hilliges O., Molyneaux D., Kim D., Davison A. J., Kohi P., Shotton J., Hodges S., Fitzgibbon A. *Mixed and augmented reality (ISMAR), 2011 10th IEEE international symposium*. 2011, pp. 127–136.
7. Kendall A., Cipolla R. *Proceedings of the International Conference on Robotics and Automation (ICRA)*. 2016, pp. 1–8.

8. Williams G., Drews P., Goldfain B., Rehg J. M., Theodorou E. A. *2016 IEEE International Conference on Robotics and Automation (ICRA)*. 2016, pp. 1433–1440.
9. Williams G., Aldrich A., Theodorou E. A. *Journal of Guidance, Control, and Dynamics*. 2017, pp. 1–14.
10. Dar'ina A. N., Diveev A. I., Prokop'ev I. V. *Voprosy teorii bezopasnosti i ustoychivosti system* [Questions of the theory of security and stability of systems]. Moscow: FITs IU RAN, 2019, iss. 21, pp. 66–77. [In Russian]
11. Weise T. *Global Optimization Algorithms – Theory and Application: Ph.D Thesis*. University of Kassel, 2008, pp. 184–186.
12. Kennedy J., Eberhart R. *Proceedings of IEEE International conference on Neural Networks*. 1995, pp. 1942–1948.
13. Diveev A., Sofronova E. *Bio-Inspired Computational Algorithms and Their Applications*. Ed. by Dr. Shangce Gao. InTech, 2012, pp. 19–42.
14. Williams G., Wagener N., Goldfain B., Drews P., Rehg J. M., Boots B., Theodorou A. E. *2017 IEEE International Conference on Robotics and Automation (ICRA) (29 May – 3 June 2017)*. 2017, pp. 361–363.

Дарьина Анна Николаевна
кандидат физико-математических наук, доцент,
ведущий научный сотрудник,
Федеральный исследовательский центр
«Информатика и управление»
Российской академии наук
(Вычислительный центр
им. А. А. Дородницына РАН)
(Россия, г. Москва, ул. Вавилова, 44, стр. 2)
E-mail: daryina@ccas.ru

Darina Anna Nikolaevna
candidate of physical and mathematical sciences,
associate professor, leading researcher,
Federal research center
«Computer science and control» of RAS
(Dorodnitsyn computer center
of the Russian Academy of Sciences)
(bld. 2, 44 Vavilova street, Moscow, Russia)

Прокопьев Игорь Витальевич
доктор технических наук,
ведущий научный сотрудник,
Федеральный исследовательский центр
«Информатика и управление»
Российской академии наук,
(Вычислительный центр
им. А. А. Дородницына РАН)
(Россия, г. Москва, ул. Вавилова, 44, стр. 2)
E-mail: fvi2014@list.ru

Prokopyev Igor Vitalevich
doctor of technical sciences, leading researcher,
Federal research center
«Computer science and control» of RAS
(Dorodnitsyn computer center
of the Russian Academy of Sciences)
(bld. 2, 44 Vavilova street, Moscow, Russia)

Образец цитирования:
Дарьина, А. Н. Метод оптимизации параметров контроллера беспилотного транспортного средства на основе оптимизации роя частиц / А. Н. Дарьина, И. В. Прокопьев // Надежность и качество сложных систем. – 2020. – № 3 (31). – С. 80–87. – DOI 10.21685/2307-4205-2020-3-10.