

**АНАЛИЗ ПРОИЗВОДИТЕЛЬНОСТИ  
И НАСТРОЙКА СТАТИЧЕСКИХ ПАРАМЕТРОВ  
ОДНОЗАПРОСНЫХ ПЛАНИРОВЩИКОВ ТРАЕКТОРИЙ**

А. ЖАРУХ, С. А. КОЛЮБИН\*

*Университет ИТМО, Санкт-Петербург, Россия*

\**s.kolyubin@itmo.ru*

**Аннотация.** Представлены результаты исследования производительности планировщиков движения, основанных на выборке одного запроса, для различной кинематики манипуляционных роботов и характеристик окружающей среды. Анализ проводился применительно к задачам поиска пути в многомерном конфигурационном пространстве. Проанализированы однонаправленные и двунаправленные планировщики по их чувствительности к изменению значений двух ключевых статических параметров: Range и Goal-Bias. Показано, что имеют место либо минимальные пороговые, либо оптимальные значения каждого из рассмотренных параметров, существенно отличающиеся для каждого алгоритма планирования, но практически инвариантные к кинематике и степени заполнения окружения робота препятствиями. Предложены рекомендации по настройке статических параметров планировщиков.

**Ключевые слова:** планирование движения, планировщики на основе выборки, кратчайший путь, манипуляционные роботы

**Ссылка для цитирования:** Жарух А., Колюбин С. А. Анализ производительности и настройка статических параметров однозапросных планировщиков траекторий // Изв. вузов. Приборостроение. 2022. Т. 65, № 5. С. 323—334. DOI: 10.17586/0021-3454-2022-65-5-323-334.

**PERFORMANCE ANALYSIS AND STATIC  
PARAMETERS SETTING OF SINGLE-QUERY TRAJECTORY PLANNERS**

A. Jaroukh, S. A. Kolyubin \*

*ITMO University, St. Petersburg, Russia*

\**s.kolyubin@itmo.ru*

**Abstract.** Results of a study of the of motion planner performance based on single query sample of a for various kinematics of robotic manipulators and environmental characteristics are presented. The analysis is carried out in relation to the problems of finding a path in a multidimensional configuration space. Unidirectional and bidirectional planners are analyzed according to their sensitivity to changes in the values of two key static parameters: Range and Goal-Bias. It is shown that there are either minimal threshold or optimal values of each of the parameters considered, which differ significantly for various planning algorithms, but are practically invariant to the kinematics and the degree of filling of the robot environment with obstacles. Recommendations on tuning the considered static parameters of the planners are proposed.

**Keywords:** motion planning, sample-based planners, shortest path, robotic manipulators

**For citation:** Jaroukh A., Kolyubin S. A. Performance analysis and static parameters setting of single-query trajectory planners. *Journal of Instrument Engineering*. 2022. Vol. 65, N 5. P. 323—334 (in Russian). DOI: 10.17586/0021-3454-2022-65-5-323-334.

**Введение.** С появлением антропоморфных роботов и коллаборативных манипуляторов с большим числом степеней свободы, зачастую избыточной кинематики, которые должны перемещаться в загроможденном пространстве избегая столкновений, выросла значимость алгоритмов планирования движения в многомерном конфигурационном пространстве сложной топологии. Одним из подходов, успешно применяемых для решения этих задач, являются планировщики движения на основе выборки, например, на основе метода быстроисследующих случайных деревьев. При этом актуальны способы повышения производительности подобных планировщиков, которая выражается в проценте успешных попыток планирования и скорости планирования, близкой к работе в реальном времени, без ухудшения оптимальности найденных решений по заданному критерию, например минимизации длины пути. Попытки обобщения характеристик планировщиков на основе выборки были предприняты, в частности, в работе [1]. Это исследование проводилось для трех манипуляторов различной кинематики в трех различных окружающих средах, чтобы дать интуитивное представление, какой планировщик в каких условиях стоит использовать.

В целом, повышение производительности планировщиков является нетривиальной задачей, так как для каждого конкретного алгоритма зависит от настройки существенного числа параметров. С другой стороны, существует этап предварительного планирования, когда могут быть определены значения параметров планировщиков. Однако этот процесс может занимать много времени, а его эффект в части повышения производительности планировщиков существенно варьируется в зависимости от специфики алгоритма.

В настоящей статье решается задача более системного анализа производительности планировщиков как функции статических параметров для разной кинематики манипуляторов, размерности конфигурационного пространства и топологии свободного рабочего пространства.

В открытой библиотеке планирования движения OMPL [2] существует 17 однозапросных планировщиков, которые интегрированы в популярный фреймворк для планирования движения роботов MoveIt! [3]. При этом для 14 из них используется параметр Range, а для 8 — два параметра: Range и Goal-Bias (смысл этих параметров подробно объясняется далее). Эти параметры известны как очень эффективные [2], поэтому именно они и рассматриваются в данной работе.

Для полноты и наглядности исследования был проведен анализ планировщиков на примере пространственных манипуляторов полной кинематики с 6 степенями свободы и избыточной кинематики с 7 степенями свободы, работающих в трех различных средах: с низким, средним и высоким уровнем заполнения пространства препятствиями. Проведенный анализ позволяет решить, какой характер зависимости производительности от вариации этих параметров имеют различные планировщики и как следует организовать процедуру настройки параметров для увеличения производительности за кратчайшее время. Более того, данное исследование закладывает основу для разработки метода и алгоритмов автоматической калибровки планировщиков, когда значимые параметры настраиваются в процессе работы.

**Обзор планировщиков движения.** Планировщики на основе выборки являются вероятностно полными, т.е. алгоритм находит по крайней мере одно выполнимое решение после достаточного количества итераций, и асимптотически оптимальными, т.е. алгоритм после длительного времени находит глобальное оптимальное решение данного запроса.

Задача глобального однозапросного планирования состоит в том, чтобы найти путь от начального состояния (конфигурации или положения центральной инструментальной точки манипулятора) в заданное целевое состояние в статичном окружении. При этом однозапросный планировщик находит решение без сохранения промежуточных состояний из предыдущих запросов.

Приведем краткую характеристику наиболее распространенных современных глобальных однозапросных планировщиков движения на основе выборки, применяемых для поиска пути в конфигурационном пространстве  $CS$ .

Один из них — планировщик на основе метода быстроисследующих случайных деревьев (Rapidly Exploring Random Trees — RRT) [4], генерирующий в свободном конфигурационном пространстве робота  $CS_{free}$  случайные точки, соответствующие определенным сочетаниям значений обобщенных координат, начиная с начального состояния в направлении неисследованных областей. Генерируемые точки (сэмплы) представляют собой узлы, которые соединяются друг с другом ребрами, образуя древовидную структуру. Добавление новых сэмплов в дерево прекращается, как только дерево достигает узла, соответствующего целевой конфигурации робота.

Известен также ряд модификаций исходного RRT-планировщика. RRTConnect [5] — это двунаправленная версия RRT, где генерируются два дерева: одно из начального состояния, а другое из целевого. На каждой итерации алгоритм проверяет, могут ли эти два дерева быть соединены, и как только это происходит, решение считается найденным. Версия планировщика RRT\* [6] добавляет шаг оптимизации на каждой итерации планирования для повторного соединения ранее сгенерированных вершин по критерию наименьших затрат. Планировщик с нижней границей LBT-RRT [7] использует коэффициент аппроксимации  $\epsilon$  при генерации дерева поиска. Его производительность варьируется между RRT\* при  $\epsilon=0$  и RRT при  $\epsilon=1$ . Оба планировщика RRT\* и LBT-RRT расходуют оставшееся допустимое время планирования на уточнение существующего решения. Трансляционное RRT (TRRT) [8] оптимизирует путь с точки зрения минимальной механической работы на соответствующее перемещение. Этот планировщик применяет тест перехода к новым сэмплам, чтобы добавить или отклонить их. Также известная двунаправленная версия планировщика TRRT — BiTRRT [9].

Кинодинамическое планирование движения путем исследования внутренних и внешних ячеек KPIECE [10] — это древовидный планировщик, оценивающий охват конфигурационного пространства с помощью слоев дискретизации на основе сетки. Такой подход позволяет уменьшить необходимый объем используемой при планировании памяти. Более того, этот планировщик выбирает новое состояние из предыдущего через управляющее воздействие, которое применяется в течение определенного времени на симуляторе робота. Такой способ планирования увеличивает точность, поскольку физические симуляторы учитывают динамические свойства робота. VKPIECE — это двунаправленная версия описанного планировщика, а LBKPIECE — тот же двунаправленный планировщик с „ленивой“ проверкой столкновений (lazy collision checking).

Расширенный пространственный планировщик EST [11], основанный на деревьях, стремится направить новые выборки в неисследованные области, которые имеют меньшую плотность выборки. ViEST — это двунаправленная версия EST, тогда как ProjEST пытается обнаружить менее исследованные области через сетку, созданную проекцией конфигурационного пространства.

Однозапросный двунаправленный вероятностный планировщик дорожных карт с ленивой проверкой столкновений SBL [12] имеет ту же стратегию расширения, что и EST. Он использует структуру данных сетки для проектирования пространства состояний, а расширение деревьев происходит в наименее заполненных ячейках сетки. Когда это возможно, планировщик соединяет два дерева без проверки того, гарантирует ли найденный путь отсутствие столкновений (ленивая проверка). Если же сгенерированный путь приводит к столкновениям, соответствующие вершины и переходы между ними будут удалены, а расширение дерева поиска продолжится.

Быстрое маршрутирующее дерево FMT [13] — асимптотически оптимальный планировщик, предназначенный для решения задач поиска пути в многомерном конфигурационном пространстве и использующий метод оптимизации „затраты-выпуск“ при генерации дерева с заданным количеством сэмплов. Планировщик BFMT представляет собой двунаправленную версию FMT.

Дерево направленного деления пути PDST [14] — это алгоритм, который осуществляет планирование посредством построения сегментов пути, а не генерирования отдельных конфигураций робота. Он обнаруживает менее исследованные области в конфигурационном пространстве, используя двоичное разбиение пространства состояний.

Поисковое дерево с независимой от разрешения оценкой плотности STRIDE [15] — это планировщик движения на основе исследующих деревьев, который обнаруживает менее изученные области конфигурационного пространства, используя структуру данных ближайших соседей GNAT.

**Анализ параметров планировщиков.** Процесс работы каждого планировщика зависит от значений ряда внутренних параметров. Часть из них относится к *динамическим параметрам*, которые автоматически настраиваются во время выполнения алгоритма; к ним, например, относится параметр температуры в планировщиках TRRT и BiTRRT [8, 9]. При этом другие параметры являются *статическими*, значения которых должны быть установлены до начала процесса планирования. Именно на анализе их влияния на работу планировщиков сосредоточено данное исследование.

Все однозапросные планировщики можно разделить на группы или семейства в зависимости от базового принципа их работы, находящего отражение в используемых уникальных параметрах: семейство KPIECE, семейство RRT, семейство EST и т.д. Как правило, для каждого семейства приводятся рекомендации по выбору некоторых подходящих значений статических параметров, однако их оптимальность для конкретного робота и его окружения остается открытым вопросом.

За исключением алгоритмов PDST, FMT и BFMT, все однозапросные планировщики имеют два общих статических параметра: Range, который выражает максимальную длину сегмента, добавляемого в дерево поиска пути, и Goal-Bias, который соответствует вероятности случайного выбора целевой конфигурации в качестве вершины дерева поиска на каждой итерации. Несмотря на то, что эти два параметра сильно влияют на производительность алгоритмов планирования [2], более детальному их анализу не было уделено достаточного внимания. Для иллюстрации на рис. 1 показано, насколько заметно изменяется плотность узлов, которые были выбраны при поиске траектории, при изменении значения всего лишь одного параметра:  $a$  — при произвольном значении Range-параметра,  $b$  — при настроенном значении Range-параметра.

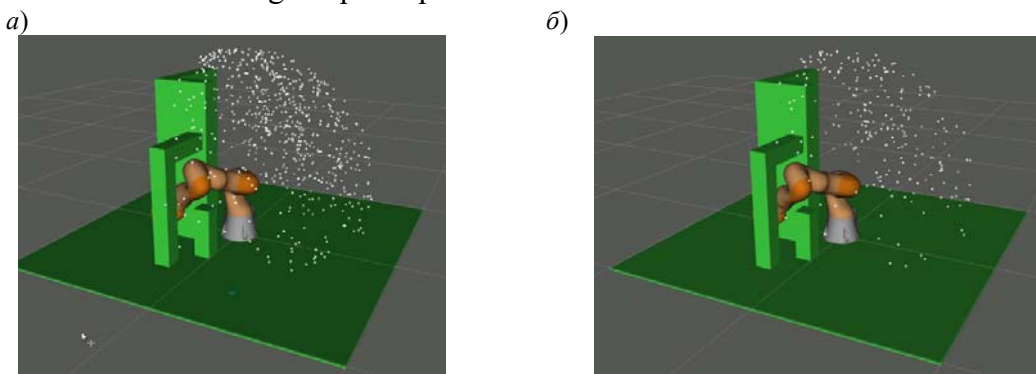


Рис. 1

В настройках OMPL [2] также есть важные *глобальные параметры*, которые необходимы для всех планировщиков, основанных на выборке. Глобальные параметры не нуждаются

в настройке, однако их значения необходимо установить в зависимости от желаемого оптимизационного критерия.

Одним из глобальных параметров является Longest-Valid-Segment-Fraction, который выражает максимально допустимую длину траектории без выполнения проверки столкновений. Необходимо подчеркнуть разницу между этим параметром и статическим параметром Range. Параметр Range можно рассматривать как максимально допустимое расстояние (длину ребра) между текущей и генерируемой вершинами в дереве поиска, а Longest-Valid-Segment-Fraction — это максимальная длина сегмента ребра (определяется как пропорция от общего размера конфигурационного пространства, в которой осуществляется поиск), при которой на промежуточных итерациях не осуществляется проверка столкновений. Увеличение значения Longest-Valid-Segment-Fraction уменьшает время планирования, но снижает безопасность найденных траекторий. В описываемых далее экспериментах для получения достоверных данных значение Longest-Valid-Segment-Fraction принято равным 0,005.

Другим важным глобальным параметром является Projection Evaluator, который используется планировщиками для оценки покрытия пространства поиска через проекцию на конфигурационное пространство более низкой размерности  $CS_{proj}$  [10—12]. Чем выше размерность  $CS_{proj}$ , тем точнее оценка покрытия  $CS$ , однако опытным путем было установлено, что время планирования увеличивается экспоненциально с ростом размерности  $CS_{proj}$ . Поэтому была выбрана размерность  $CS_{proj}$ , равная двум, и задан параметр Projection Evaluator на первое и второе сочленение (рассматривается робот артикулированной кинематики).

Наконец, глобальный параметр Optimization Objective определяет критерий оптимизации для планировщика. Этот параметр был установлен в значение Mechanical-Work-Optimization-Objective — минимизация затрат механической энергии на перемещение, так как он одновременно с минимизацией длины траектории приводит к генерации более естественных траекторий [9].

**Экспериментальные исследования. Робот-манипулятор.** Производительность разных планировщиков была проанализирована на примере задачи планирования движения коллаборативного манипулятора последовательной кинематики KUKA LBR IIWA 14 R820 [16]; его модель с обозначением осей вращения показана на рис. 2.

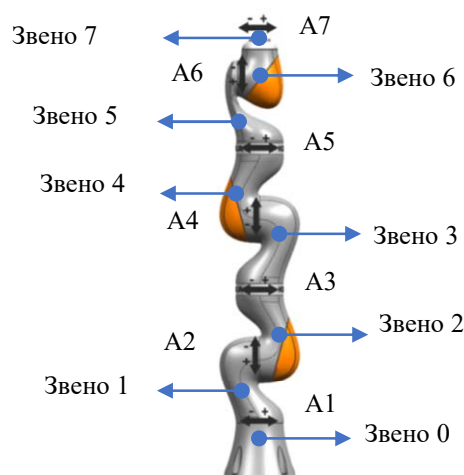


Рис. 2

Рассматривались два варианта:

— избыточная кинематика, когда в движение приводятся все 7 сочленений робота и, соответственно, размерность конфигурационного пространства, в котором осуществляется планирование, равна 7;

— полная кинематика с той же геометрией робота и движением только по 6 осям; вариант полной кинематики был получен путем ограничения движения („заморозки“) третьего сочленения (А3) робота.

*Характеристики окружающей среды.* Чтобы выяснить, как изменение окружения влияет на выбор оптимальных значений параметров планировщиков, в ходе исследования были использованы три типа окружения: слабо, средне и сильно загроможденные среды. Слабо загроможденная среда характеризуется тем, что манипулятор имеет свободное пространство вокруг начального положения и загроможденное пространство (заполненное объектами, с которыми необходимо избежать столкновения) около целевого положения. Эта модель окружения показана на рис. 3, а (3D-представление), а топология свободного пространства (без столкновений) в проекции относительно обобщенных координат первого, второго и четвертого сочленений ( $\theta_1, \theta_2, \theta_4$  — талии, плеча и локтя) — на рис. 3, б и в. Подобная проекция позволяет представить изменение конфигураций манипулятора как движение точки в трехмерном пространстве. Плотность свободного пространства  $CS_{\text{free}}$  по этой проекции составляет 84 %. Напомним, что поставленная задача планирования движения решается в конфигурационном пространстве размерности 7 или 6, а приведенная проекция — лишь способ доступной визуализации.

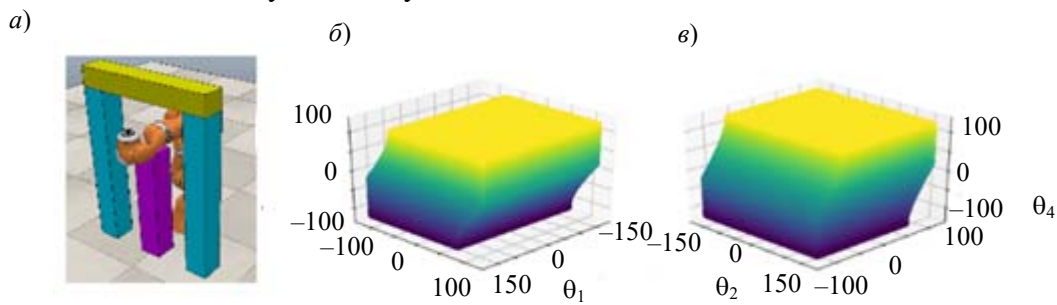


Рис. 3

Средне загроможденная среда характеризуется тем, что и при начальном, и при целевом положениях имеются в равной степени заполненные препятствиями пространства, а в средней области между двумя крайними положениями имеется свободное связное подпространство. Это означает, что робот будет частично свободен для перемещения в некоторых областях своего конфигурационного пространства. На рис. 4, а показана модель такой среды, а топология свободного пространства в проекции относительно обобщенных координат первого, второго и четвертого сочленений показана на рис. 4, б и в. Плотность свободного пространства  $CS_{\text{free}}$  в этой проекции составляет 37 %.

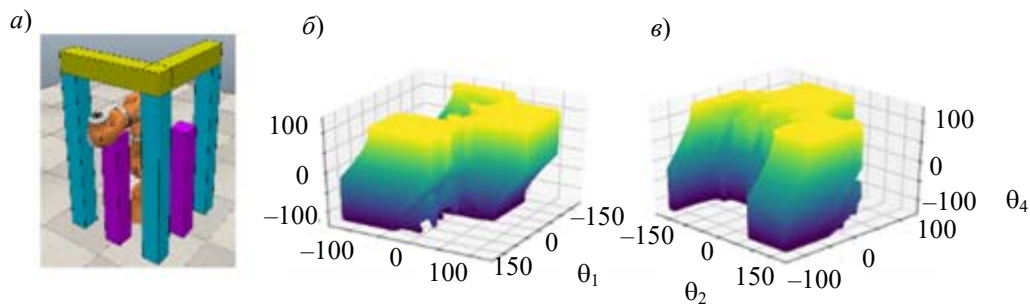


Рис. 4

В сильно загроможденной среде препятствия имеются на всем пути от начального положения до целевого. В этой модели отсутствуют большие участки связного конфигурационного подпространства, где новые сэмплы могут генерироваться и свободно соединяться

друг с другом без риска столкновений. Эта модель показана на рис. 5, а, а топология свободного пространства в проекции относительно обобщенных координат первого, второго и четвертого сочленений — на рис. 5, б и в. Плотность свободного пространства  $CS_{\text{free}}$  в этой проекции составляет всего 8 %.

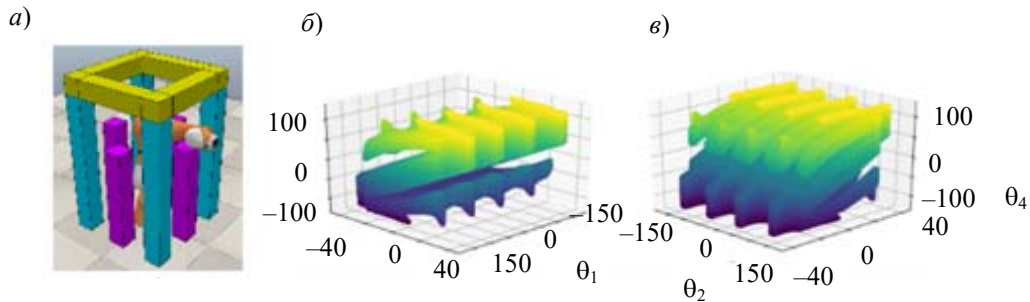


Рис. 5

**Программное обеспечение.** Исследование было выполнено с помощью программного обеспечения с открытым исходным кодом Robot Operating System [17] с использованием фреймворка MoveIt! и алгоритмов из открытой библиотеки OMPL. В MoveIt! решение, найденное планировщиком из OMPL, далее сглаживается с помощью стандартного процесса упрощения пути.

Эксперименты проводились в симуляторе RViz с использованием однопоточных вычислений на персональном компьютере с процессором Intel i5 2.3 ГГц и 8 Гбайт оперативной памяти. Максимально допустимое время планирования для всех запусков было установлено на 10 с.

**Оценка производительности планировщиков.** Для каждого планировщика и двух различных вариантов кинематики робота в каждой из трех predetermined сред при каждом фиксированном значении анализируемого параметра (Range и Goal-Bias) было выполнено по 100 запусков алгоритма планирования движения. На основе полученных данных были рассчитаны средние значения метрик производительности: времени планирования и длины пути для успешных запусков, а также процент таковых.

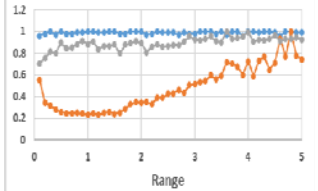
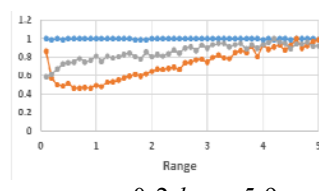
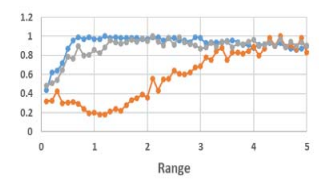
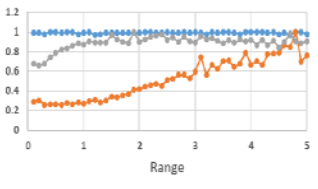
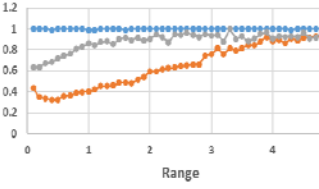
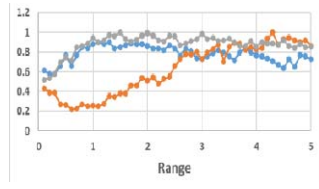
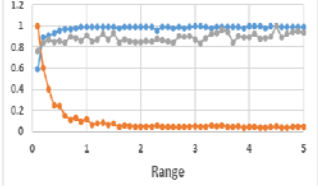
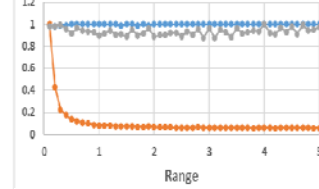
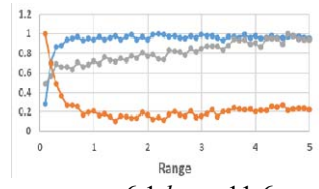
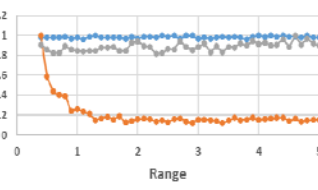
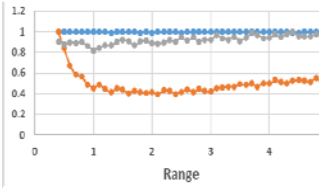
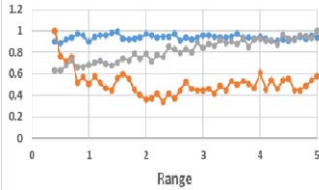
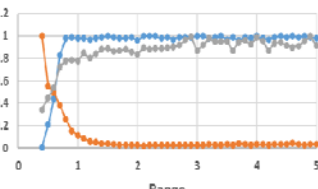
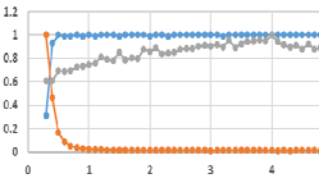
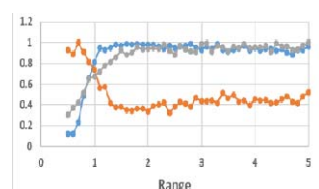
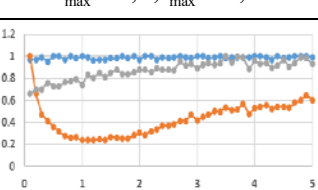
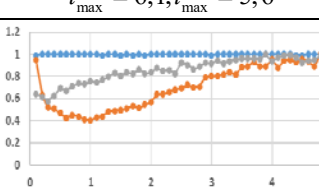
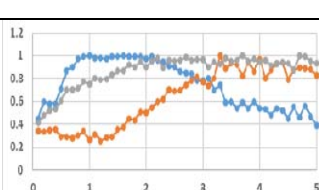
Параметр Range варьировался в диапазоне  $[0,1 \rightarrow 5]$  с шагом 0,1. Значения выше 5 были также протестированы, но ни для одного из планировщиков не показали изменения производительности, поэтому не представлены в данной работе. Параметр Goal-Bias варьировался в трех различных диапазонах:  $[0,005 \rightarrow 0,25]$  с шагом 0,005, затем  $[0,3 \rightarrow 0,5]$  с шагом 0,02, и наконец  $[0,5 \rightarrow 1,0]$  с шагом 0,1.

**Полученные результаты.** Результаты оценки производительности планировщиков как функции от параметров Range и Goal-Bias представлены в табл. 1 и 2 соответственно, где оранжевый цвет — время планирования ( $t$ ), серый — длина траектории ( $l$ ), синий — доля удачных запусков. Чтобы показать зависимости для трех различных показателей на одном рисунке, графики были нормированы по максимальному значению (эти значения приведены на графиках в таблицах). Доля успешных запусков измерена в диапазоне от 0 до 1 (100 %).

**Range-параметр.** Табл. 1 иллюстрирует изменение производительности планировщиков в зависимости от параметра Range для манипулятора полной и избыточной кинематики в средне и сильно загроможденных средах. Приведены результаты только для двунаправленных планировщиков, поскольку однонаправленные показали несистематические результаты, так что процесс настройки параметра Range для них не приводит к значительному повышению производительности.

Как видно из графиков, при низких значениях Range все планировщики работают относительно плохо, так как низкие значения заставляют дерево выборов двигаться медленно (маленькими шагами), когда оно исследует пространство состояний.

Таблица 1

Алгоритм планирования	Избыточная кинематика, средне загроможденная среда	Полная кинематика, средне загроможденная среда	Избыточная кинематика, сильно загроможденная среда
ВКPIECE	 $t_{\max} = 0,6; l_{\max} = 8,8$	 $t_{\max} = 0,2; l_{\max} = 5,9$	 $t_{\max} = 4; l_{\max} = 12,7$
LBKPIECE	 $t_{\max} = 0,8; l_{\max} = 9,4$	 $t_{\max} = 0,3; l_{\max} = 5,9$	 $t_{\max} = 5,3; l_{\max} = 11,4$
RRTConnect	 $t_{\max} = 4,5; l_{\max} = 8,3$	 $t_{\max} = 1,1; l_{\max} = 4,6$	 $t_{\max} = 6,1; l_{\max} = 11,6$
BiTRRT	 $t_{\max} = 1,9; l_{\max} = 8,8$	 $t_{\max} = 0,3; l_{\max} = 5,4$	 $t_{\max} = 2,7; l_{\max} = 12,6$
BiEST	 $t_{\max} = 1,9; l_{\max} = 8,8$	 $t_{\max} = 6,1; l_{\max} = 5,6$	 $t_{\max} = 5,6; l_{\max} = 11,5$
SBL	 $t_{\max} = 0,7; l_{\max} = 8,8$	 $t_{\max} = 0,2; l_{\max} = 5,7$	 $t_{\max} = 7,2; l_{\max} = 11,9$

Планировщики, использующие-grid-структуру данных для проектирования пространства состояний (ВКPIECE, LBKPIECE, SBL, BiEST), как правило, имеют оптимальное значение Range (глобальный оптимум), при котором производительность планировщика характеризуется идеальным сочетанием доли успешных запусков, времени планирования и длины траектории. При высоких значениях Range время планирования заметно увеличивается, что, вероятно, связано с тем, что в этом случае планировщики проходят дальше (к границе конфигурационного пространства) в поисках областей с низкой плотностью выборки, и это может замедлять соеди-



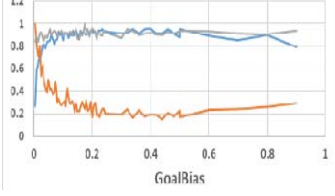
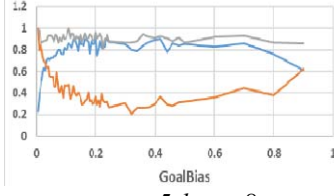
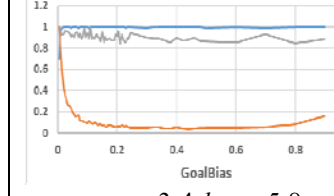
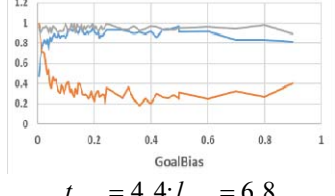
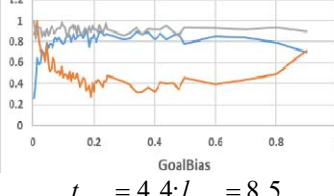
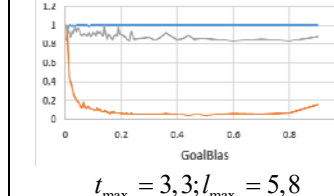
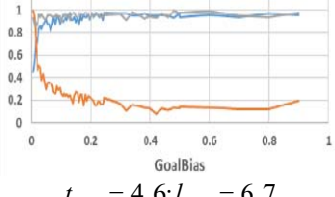
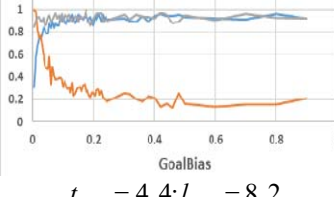
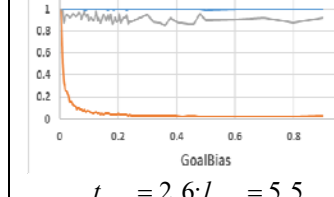
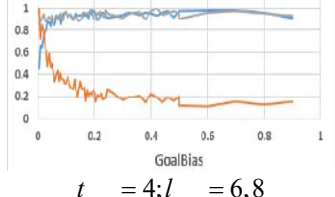
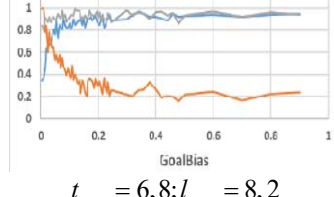
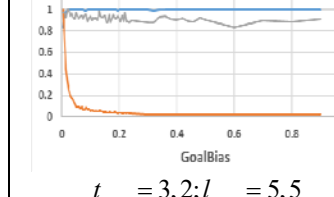
нение двунаправленных деревьев. Также планировщики SBL и LBKPIECE используют ленивую проверку столкновений (проводится только после построения полного пути), поэтому в сильно загроможденной среде большие шаги поиска (большие значения Range) приводят к тому, что существенная доля найденных планировщиком промежуточных решений впоследствии будет исключена.

RRTConnect и BiTRRT имеют одну цель — соединить два дерева (в дополнение к шагу оптимизации в BiTRRT). Напомним, что при генерации сэмплов добавляются промежуточные вершины между текущей крайней вершиной дерева и следующей случайной в пределах шага, задаваемого параметром Longest-Valid-Segment-Fraction. Благодаря этой методике можно предположить, что после порогового значения Range производительность планировщика не изменится, и два двунаправленных дерева будут находить путь друг к другу одинаковым образом.

Следует отметить, согласно анализу табл. 1, что изменение среды не влияет на оптимальное значение параметра Range. Более того, редуцирование числа степеней подвижности манипулятора (меньшая размерность конфигурационного пространства) не требует перенастройки этого параметра.

*Параметр Goal-Bias.* В табл. 2 представлены результаты работы планировщиков в зависимости от параметра Goal-Bias, когда задача планирования движения применялась к манипулятору избыточной кинематики в средах с низкой и средней плотностью препятствий, а также к манипулятору полной кинематики в средне загроможденной среде. Приведены результаты для однонаправленных планировщиков, за исключением RRT, TRRT, RRT\* и LBTRRT, поскольку эти планировщики показали несистематические результаты, так что настройка параметра Goal-Bias не приводит к значительному повышению производительности.

Таблица 2

Алгоритм планирования	Избыточная кинематика, слабо загроможденная среда	Избыточная кинематика, средне загроможденная среда	Полная кинематика, средне загроможденная среда
KPIECE	 $t_{\max} = 4,5; l_{\max} = 6,6$	 $t_{\max} = 5,5; l_{\max} = 8$	 $t_{\max} = 3,4; l_{\max} = 5,8$
STRIDE	 $t_{\max} = 4,4; l_{\max} = 6,8$	 $t_{\max} = 4,4; l_{\max} = 8,5$	 $t_{\max} = 3,3; l_{\max} = 5,8$
EST	 $t_{\max} = 4,6; l_{\max} = 6,7$	 $t_{\max} = 4,4; l_{\max} = 8,2$	 $t_{\max} = 2,6; l_{\max} = 5,5$
ProjEST	 $t_{\max} = 4; l_{\max} = 6,8$	 $t_{\max} = 6,8; l_{\max} = 8,2$	 $t_{\max} = 3,2; l_{\max} = 5,5$

Анализ производительности по параметру Goal-Bias показывает, что целесообразно половину случайных сэмплов генерировать в целевом положении в конфигурационном пространстве (Goal-Bias = 0,5). Это не согласуется со значением Goal-Bias  $\approx 0,05$ , как рекомендуется в [2]. Разница заключается в том, что для манипуляторов с фиксированной базой пространство поиска пути значительно меньше, чем для движущегося в лабиринте мобильного робота, рассматриваемого в [2].

**Заключение.** Проанализированы характеристики планировщиков движения, основанных на выборке одного запроса, в зависимости от параметров Range и Goal-Bias с использованием манипулятора ПWA в избыточной (7 степеней свободы) и в полной (6 степеней свободы) кинематике. Построены графики эффективности планирования, времени планирования и длины пути при изменении значений параметров в заданных диапазонах. Каждая точка на графиках определялась как среднее значение соответствующей величины на основе 100 запусков.

Анализ зависимости производительности планировщиков от параметра Range показал следующее:

- настройка параметра Range имеет наибольшее значение для двунаправленных планировщиков (RRTConnect, BiTRRT, (L)BKPIECE, BiEST, SBL);
- для всех проанализированных планировщиков оптимальные значения Range-параметра инварианты к степени загроможденности окружения робота;
- для планировщиков на основе сетки, использующих функцию sample-near ((L)BKPIECE, SBL), оптимальное значение Range-параметра располагается так, что увеличение значения, равно как и уменьшение, приводит к существенному росту времени на планирование; для других планировщиков (RRTConnect, BiTRRT, BiEST) с равномерным распределением генерируемых сэмплов (uniform sampling) имеется некоторое граничное значение Range-параметра, так что при уменьшении Range от него время планирования растет, а при увеличении от него — не изменяется;
- для планировщиков с ленивой проверкой столкновений (LBKPIECE и SBL) увеличение Range-параметра приводит к уменьшению числа успешных попыток планирования;
- сравнение производительности планировщиков для полной и избыточной кинематики манипулятора показало, что редуцирование количества приводимых в движение сочленений (заморозка) не требует перенастройки Range-параметра.

Анализ зависимости производительности мононаправленных планировщиков от параметра Goal-Bias показал следующее:

- для манипуляционных роботов, как правило, имеющих пространство поиска (конфигурационное) небольшого размера, рекомендованное OMPL значение Goal-Bias 0,05 [2] является плохим выбором; для большинства проанализированных планировщиков оптимальным можно считать значение Goal-Bias 0,5 и выше;
- выбор высокого значения параметра Goal-Bias целесообразен для манипуляторов, где существует высокая вероятность наличия свободного от столкновений пути (в конфигурационном пространстве), соединяющего начальное и целевое положения робота.

Данная работа создает задел для будущих исследований, направленных на интеграцию автоматического метода настройки планировщиков, в котором параметры подбираются онлайн в процессе планирования.

#### СПИСОК ЛИТЕРАТУРЫ

1. Meijer J., Lei Q., Wisse M. Performance study of single-query motion planning for grasp execution using various manipulators // 18th Intern. Conf. on Advanced Robotics (ICAR). 2017. P. 450—457.
2. Sucan I. A., Moll M., Kavraki L. E. The open motion planning library // IEEE Robot. Autom. Mag., 2012. Vol. 19, N 4. P. 72—82.

3. *Sucan I. A., Chitta S.* Moveit!. *IEEE Robotics & Automation Magazine*. 2012. March, N 1(19). P. 18—19. DOI: 10.1109/MRA.2011.2181749.
4. *LaValle S. M.* *Rapidly-Exploring Random Trees: A New Tool for Path Planning*. Berlin: Springer-Verlag, 1998.
5. *Kuffner J. J., LaValle S. M.* RRT-connect: An efficient approach to single-query path planning // *Proc. ICRA, Millennium Conf. IEEE Intern. Conf. on Robotics and Automation*. 2000. Vol. 2. P. 995—1001.
6. *Karaman S., Frazzoli E.* Sampling-based algorithms for optimal motion planning // *Intern. Journal Robotics Research*. 2011. Vol. 30, N 7. P. 846—894.
7. *Salzman O., Halperin D.* Asymptotically near-optimal RRT for fast, high-quality motion planning // *IEEE Trans. Robot.* 2016. P. 473—483.
8. *Jaillet L., Cortés J., Siméon T.* Sampling-based path planning on configuration-space costmaps // *IEEE Trans. Robot.* 2010. Vol. 26, N 4. P. 635—646.
9. *Devaurs D., Siméon T., Cortés J.* Enhancing the transition-based RRT to deal with complex cost spaces // *IEEE Intern. Conf. on Robotics and Automation*. 2013. P. 4120—4125.
10. *Şucan I. A., Kavraki L. E.* Kinodynamic motion planning by interior-exterior cell exploration // *Algorithmic Foundation of Robotics VIII*. Springer, 2009. P. 449—464.
11. *Hsu D., Latombe J.-C., Motwani R.* Path planning in expansive configuration spaces // *Proc. of Intern. Conf. on Robotics and Automation*. 1997. Vol. 3. P. 2719—2726.
12. *Sánchez G., Latombe J.-C.* A single-query bi-directional probabilistic roadmap planner with lazy collision checking // *Robotics Research*. Springer, 2003. P. 403—417.
13. *Janson L., Schmerling E., Clark A., Pavone M.* Fast marching tree: A fast marching sampling-based method for optimal motion planning in many dimensions // *Intern. Journal Robotics Research*. 2015. Vol. 34, N 7. P. 883—921.
14. *Ladd A. M., Kavraki L. E.* Motion Planning in the Presence of Drift, Underactuation and Discrete System Changes // *Robotics: Science and Systems*. 2005. Vol. 1. P. 233—240.
15. *Gipson B., Moll M., Kavraki L. E.* Resolution independent density estimation for motion planning in high-dimensional spaces // *IEEE Intern. Conf. on Robotics and Automation*. 2013. P. 2437—2443.
16. KUKA, KUKA LBR [Электронный ресурс]: <<http://www.kuka.com/>>, 2021-04-10.
17. *Quigley M., Conley K., Gerkey B., Faust J., Foote T., Leibs J., Wheeler R., Ng A. Y.* ROS: an open-source Robot Operating System // *ICRA Workshop on Open-Source Software*. 2009. Vol. 3, N 3.2. P. 5.

#### Сведения об авторах

- Абдулраззак Жарух** — аспирант; Университет ИТМО, факультет систем управления и робототехники; E-mail: [abdelrazaq.ja@gmail.com](mailto:abdelrazaq.ja@gmail.com)
- Сергей Алексеевич Колюбин** — д-р техн. наук, доцент; Университет ИТМО, факультет систем управления и робототехники, лаборатория биомехатроники и энергоэффективной робототехники; вед. научный сотрудник; E-mail: [s.kolyubin@itmo.ru](mailto:s.kolyubin@itmo.ru)

Поступила в редакцию 18.02.22; одобрена после рецензирования 03.03.22; принята к публикации 29.03.22.

#### REFERENCES

1. Meijer J., Lei Q., and Wisse M. *2017 18th International Conference on Advanced Robotics (ICAR)*, 2017, pp. 450–457.
2. Sucan I.A., Moll M., and Kavraki L.E. *IEEE Robot. Autom. Mag.*, 2012, no. 4(19), pp. 72–82.
3. Sucan I.A. and Chitta S., Cousins S. *IEEE Robotics & Automation Magazine*, 2012, March, no. 1(19), pp. 18–19, DOI:10.1109/MRA.2011.2181749.
4. LaValle S.M. *Rapidly-exploring random trees: A new tool for path planning*, Report No. TR 98-11, 1998.
5. Kuffner J.J. and LaValle S.M. *Proceedings 2000 ICRA. Millennium Conference, IEEE International Conference on Robotics and Automation, Symposia Proceedings (Cat. No. 00CH37065)*, 2000, vol. 2, pp. 995–1001.
6. Karaman S. and Frazzoli E. *Int. J. Rob. Res.*, 2011, no. 7(30), pp. 846–894.
7. Salzman O. and Halperin D. *IEEE Trans. Robot.*, 2016, pp. 473–483.
8. Jaillet L., Cortés J., and Siméon T. *IEEE Trans. Robot.*, 2010, no. 4(26), pp. 635–646.
9. Devaurs D., Siméon T., and Cortés J. *2013 IEEE International Conference on Robotics and Automation*, 2013, pp. 4120–4125.

10. Şucan I.A. and Kavraki L.E. *Algorithmic Foundation of Robotics VIII*, Springer, 2009, pp. 449–464.
11. Hsu D., Latombe J.-C., and Motwani R. *Proceedings of International Conference on Robotics and Automation*, 1997, vol. 3, pp. 2719–2726.
12. Sánchez G. and Latombe J.-C. *Robotics Research*, Springer, 2003, pp. 403–417.
13. Janson L., Schmerling E., Clark A., and Pavone M. *Int. J. Rob. Res.*, 2015, no. 7(34), pp. 883–921.
14. Ladd A.M. and Kavraki L.E. *Robotics: Science and Systems*, 2005, vol. 1, pp. 233–240.
15. Gipson B., Moll M., and Kavraki L.E. *2013 IEEE International Conference on Robotics and Automation*, 2013, pp. 2437–2443.
16. *KUKA*, <http://www.kuka.com/>.
17. Quigley M., Conley K., Gerkey B., Faust J., Foote T., Leibs J., Wheeler R., and Ng, A.Y. *ICRA Workshop on Open-Source Software*, 2009, May, no. 2(3), pp. 5.

**Data on authors**

- Abdulrazzak Jaroukh** — Post-Graduate Student; ITMO University, Faculty of Control Systems and Robotics; E-mail: abdelrazaq.ja@gmail.com
- Sergey A. Kolyubin** — Dr. Sci., Associate Professor; ITMO University, Faculty of Control Systems and Robotics, Laboratory of Biomechatronics and Energy-Efficient Robotics; Leading Scientist; E-mail: s.kolyubin@itmo.ru

Received 18.02.22; approved after reviewing 03.03.22; accepted for publication 29.03.22.